BERT on SQUAD Dataset

- Most of the code is taken referance from google-research
- The bert model is **fine-tuned** only.
- Used the small bert model
- All the referance are mentioned in the referances section

In [0]:
```
# importing all necessary files

import zipfile
from matplotlib import pyplot as plt
%matplotlib inline
import sys
import datetime
import tensorflow as tf
import os
import json
import six

from collections import Counter
import string
import re
import argparse
import sys
```

Get Model Files

In [1]:
```
import datetime
import json
import os
import pprint
import random
import string
import sys
import tensorflow as tf

assert 'COLAB_TPU_ADDR' in os.environ, 'ERROR: Not connected to a TPU runtime;
please see the first cell in this notebook for instructions!'
TPU_ADDRESS = 'grpc://' + os.environ['COLAB_TPU_ADDR']
print('TPU address is', TPU_ADDRESS)

from google.colab import auth
auth.authenticate_user()
with tf.Session(TPU_ADDRESS) as session:
  print('TPU devices:')
  pprint.pprint(session.list_devices())

  # Upload credentials to TPU.
  with open('/content/adc.json', 'r') as f:
    auth_info = json.load(f)
  tf.contrib.cloud.configure_gcs(session, credentials=auth_info)
  # Now credentials are set for all future sessions on this TPU.
```

```
TPU address is grpc://10.69.74.146:8470

WARNING: Logging before flag parsing goes to stderr.
W0825 16:07:45.187536 140319166060416 lazy_loader.py:50]
The TensorFlow contrib module will not be included in TensorFlow 2.0.
For more information, please see:
  * https://github.com/tensorflow/community/blob/master/rfcs/20180907-contrib
-sunset.md
  * https://github.com/tensorflow/addons
  * https://github.com/tensorflow/io (for I/O related ops)
If you depend on functionality not listed there, please file an issue.


TPU devices:
[_DeviceAttributes(/job:tpu_worker/replica:0/task:0/device:CPU:0, CPU, -1, 18
165515893288348800),
 _DeviceAttributes(/job:tpu_worker/replica:0/task:0/device:XLA_CPU:0, XLA_CP
U, 17179869184, 8778042299146624752),
 _DeviceAttributes(/job:tpu_worker/replica:0/task:0/device:TPU:0, TPU, 171798
69184, 5262853629327189391),
 _DeviceAttributes(/job:tpu_worker/replica:0/task:0/device:TPU:1, TPU, 171798
69184, 15522775392437584893),
 _DeviceAttributes(/job:tpu_worker/replica:0/task:0/device:TPU:2, TPU, 171798
69184, 863550791859807253),
 _DeviceAttributes(/job:tpu_worker/replica:0/task:0/device:TPU:3, TPU, 171798
69184, 15967457439584850702),
 _DeviceAttributes(/job:tpu_worker/replica:0/task:0/device:TPU:4, TPU, 171798
69184, 9825514860354380673),
 _DeviceAttributes(/job:tpu_worker/replica:0/task:0/device:TPU:5, TPU, 171798
69184, 9799457881871349772),
 _DeviceAttributes(/job:tpu_worker/replica:0/task:0/device:TPU:6, TPU, 171798
69184, 17961220308526691231),
 _DeviceAttributes(/job:tpu_worker/replica:0/task:0/device:TPU:7, TPU, 171798
69184, 4856695259171467612),
 _DeviceAttributes(/job:tpu_worker/replica:0/task:0/device:TPU_SYSTEM:0, TPU_
SYSTEM, 8589934592, 13923937947632350643)]
```

In [2]:
```
import sys

!test -d bert_repo || git clone https://github.com/google-research/bert bert_r
epo
if not 'bert_repo' in sys.path:
  sys.path += ['bert_repo']

# import python modules defined by BERT
import run_squad
import modeling
import optimization
import tokenization
import tensorflow as tf
import tokenization

# import tfhub
import tensorflow_hub as hub
```

```
Cloning into 'bert_repo'...
remote: Enumerating objects: 333, done.
remote: Total 333 (delta 0), reused 0 (delta 0), pack-reused 333
Receiving objects: 100% (333/333), 282.46 KiB | 3.82 MiB/s, done.
Resolving deltas: 100% (183/183), done.

W0825 16:07:52.373191 140319166060416 deprecation_wrapper.py:119] From bert_r
epo/optimization.py:87: The name tf.train.Optimizer is deprecated. Please use
tf.compat.v1.train.Optimizer instead.
```

In [3]:
```
TASK = 'SQUAD' #@param {type:"string"}
assert TASK in ('MRPC', 'CoLA',"SQUAD"), 'Only (MRPC, CoLA) are demonstrated h
ere.'


BUCKET = 'bert-on-squad' #@param {type:"string"}
assert BUCKET, 'Must specify an existing GCS bucket name'
OUTPUT_DIR = 'gs://{}/bert-tfhub/models/{}'.format(BUCKET, TASK)
tf.gfile.MakeDirs(OUTPUT_DIR)
print('***** Model output directory: {} *****'.format(OUTPUT_DIR))

# Available pretrained model checkpoints:
#   uncased_L-12_H-768_A-12: uncased BERT base model
#   uncased_L-24_H-1024_A-16: uncased BERT large model
#   cased_L-12_H-768_A-12: cased BERT large model
BERT_MODEL = 'uncased_L-12_H-768_A-12' #@param {type:"string"}
BERT_MODEL_HUB = 'https://tfhub.dev/google/bert_' + BERT_MODEL + '/1'
```

```
***** Model output directory: gs://bert-on-squad/bert-tfhub/models/SQUAD ****
*
```

In [0]:
```
# Setup TPU related config
tpu_cluster_resolver = tf.contrib.cluster_resolver.TPUClusterResolver(TPU_ADDR
ESS)
NUM_TPU_CORES = 8
ITERATIONS_PER_LOOP = 1000
```

In [5]:
```
# Setup task specific model and TPU running config.
BERT_PRETRAINED_DIR = 'gs://cloud-tpu-checkpoints/bert/' + BERT_MODEL
print('***** BERT pretrained directory: {} *****'.format(BERT_PRETRAINED_DIR))
!gsutil ls $BERT_PRETRAINED_DIR
```

```
***** BERT pretrained directory: gs://cloud-tpu-checkpoints/bert/uncased_L-12
_H-768_A-12 *****
gs://cloud-tpu-checkpoints/bert/uncased_L-12_H-768_A-12/bert_config.json
gs://cloud-tpu-checkpoints/bert/uncased_L-12_H-768_A-12/bert_model.ckpt.data-
00000-of-00001
gs://cloud-tpu-checkpoints/bert/uncased_L-12_H-768_A-12/bert_model.ckpt.index
gs://cloud-tpu-checkpoints/bert/uncased_L-12_H-768_A-12/bert_model.ckpt.meta
gs://cloud-tpu-checkpoints/bert/uncased_L-12_H-768_A-12/checkpoint
gs://cloud-tpu-checkpoints/bert/uncased_L-12_H-768_A-12/vocab.txt
```

In [0]:
```
CONFIG_FILE = os.path.join(BERT_PRETRAINED_DIR, 'bert_config.json')
INIT_CHECKPOINT = os.path.join(BERT_PRETRAINED_DIR, 'bert_model.ckpt')
VOCAB_FILE = os.path.join(BERT_PRETRAINED_DIR, 'vocab.txt')
DO_LOWER_CASE = BERT_MODEL.startswith('uncased')
```

In [0]:
```
OUTPUT_DIR = OUTPUT_DIR.replace('bert-tfhub', 'bert-checkpoints')
tf.gfile.MakeDirs(OUTPUT_DIR)
```

Get SQUAD Dataset

In [9]:
```
! wget wget --header="Host: storage.googleapis.com" --header="User-Agent: Mozi
lla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
hrome/76.0.3809.100 Safari/537.36" --header="Accept: text/html,application/xht
ml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signe
d-exchange;v=b3" --header="Accept-Language: en-US,en;q=0.9" --header="Referer:
https://www.kaggle.com/" "https://storage.googleapis.com/kaggle-datasets/374/7
81/dev-v1.1.json.zip?GoogleAccessId=web-data@kaggle-161607.iam.gserviceaccoun
t.com&Expires=1566788992&Signature=oAc8WODIx9PK4G6fnjyWSOCNZ0fSFB0J5WX%2B%2B96
gACb9Lo3QNhD4%2F3WyADcZvaTL11cCsHFk8I8vFCpdvpUetT1o6HovYGm2L3eoJSeRVPfld5LMBCX
eVrfdUXJJpX8nyDscXTMcATx4vdPulTUBaU5kZDl%2FUxwRKUqSNBo8PXp7%2Bi2BZ3CXyxLAwelaP
J5DeaadTnIXVK%2BNgaWiijMOCKmDEzGX805hvXUi6jV8eK6tgqIcy35FX2ZxzUBeUXpFozo7iCSbq
RH352KyndzoxdCCLGL65MTBBzHuvyR3hDy01repSlQItffCEh4IYjptJtYHOlQxHT7np55qXz7mvw%
3D%3D" -O "dev-v1.1.json.zip" -c
! wget wget --header="Host: storage.googleapis.com" --header="User-Agent: Mozi
lla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
hrome/76.0.3809.100 Safari/537.36" --header="Accept: text/html,application/xht
ml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signe
d-exchange;v=b3" --header="Accept-Language: en-US,en;q=0.9" --header="Referer:
https://www.kaggle.com/" "https://storage.googleapis.com/kaggle-datasets/374/7
81/train-v1.1.json.zip?GoogleAccessId=web-data@kaggle-161607.iam.gserviceaccou
nt.com&Expires=1566789042&Signature=NZjKto8MWgK2CM3Jt3nfZldBNVGyXJMJrYGinSyyO%
2B0GbznnbCJ6XN9wVSaOJKwxfCHasY%2B2A%2FPRpXBV9wJ3fPJBp3G0xHEWGEgIRt1mMt7G%2FJkY
hey94%2FRGMawt0Q9Gu4KHP%2BWoBGV6xStD9RLZyFVQZD9naImwKRD6WZBuZIdz0G6nh6gliCAhq7
LVF4cGRKxEoHWIwO91cwrfWH5KLVjbjUUMVhht2qfH0jgYI1kRTxtEVxjMZ%2Bu85GGPsWhgos2kAe
qxCF0jmZY8a%2BwgMOAXlx6fQ%2B6Fma4RyQQX6GB5hhcEaEmUs1E2awlT6sF6j3PZlc3xY4fTtVm8
PhLtqQ%3D%3D" -O "train-v1.1.json.zip" -c
```

```
--2019-08-25 16:08:23--  http://wget/
Resolving wget (wget)... failed: Name or service not known.
wget: unable to resolve host address ‘wget’
--2019-08-25 16:08:23--  https://storage.googleapis.com/kaggle-datasets/374/7
81/dev-v1.1.json.zip?GoogleAccessId=web-data@kaggle-161607.iam.gserviceaccoun
t.com&Expires=1566788992&Signature=oAc8WODIx9PK4G6fnjyWSOCNZ0fSFB0J5WX%2B%2B9
6gACb9Lo3QNhD4%2F3WyADcZvaTL11cCsHFk8I8vFCpdvpUetT1o6HovYGm2L3eoJSeRVPfld5LMB
CXeVrfdUXJJpX8nyDscXTMcATx4vdPulTUBaU5kZDl%2FUxwRKUqSNBo8PXp7%2Bi2BZ3CXyxLAwe
laPJ5DeaadTnIXVK%2BNgaWiijMOCKmDEzGX805hvXUi6jV8eK6tgqIcy35FX2ZxzUBeUXpFozo7i
CSbqRH352KyndzoxdCCLGL65MTBBzHuvyR3hDy01repSlQItffCEh4IYjptJtYHOlQxHT7np55qXz
7mvw%3D%3D
Resolving storage.googleapis.com (storage.googleapis.com)... 172.217.212.128,
2607:f8b0:4001:c05::80
Connecting to storage.googleapis.com (storage.googleapis.com)|172.217.212.128
|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1092630 (1.0M) [application/zip]
Saving to: ‘dev-v1.1.json.zip’

dev-v1.1.json.zip   100%[===================>]   1.04M  --.-KB/s    in 0.007s

2019-08-25 16:08:23 (144 MB/s) - ‘dev-v1.1.json.zip’ saved [1092630/1092630]

FINISHED --2019-08-25 16:08:23--
Total wall clock time: 0.2s
Downloaded: 1 files, 1.0M in 0.007s (144 MB/s)
--2019-08-25 16:08:26--  http://wget/
Resolving wget (wget)... failed: Name or service not known.
wget: unable to resolve host address ‘wget’
--2019-08-25 16:08:26--  https://storage.googleapis.com/kaggle-datasets/374/7
81/train-v1.1.json.zip?GoogleAccessId=web-data@kaggle-161607.iam.gserviceacco
unt.com&Expires=1566789042&Signature=NZjKto8MWgK2CM3Jt3nfZldBNVGyXJMJrYGinSyy
O%2B0GbznnbCJ6XN9wVSaOJKwxfCHasY%2B2A%2FPRpXBV9wJ3fPJBp3G0xHEWGEgIRt1mMt7G%2F
JkYhey94%2FRGMawt0Q9Gu4KHP%2BWoBGV6xStD9RLZyFVQZD9naImwKRD6WZBuZIdz0G6nh6gliC
Ahq7LVF4cGRKxEoHWIwO91cwrfWH5KLVjbjUUMVhht2qfH0jgYI1kRTxtEVxjMZ%2Bu85GGPsWhgo
s2kAeqxCF0jmZY8a%2BwgMOAXlx6fQ%2B6Fma4RyQQX6GB5hhcEaEmUs1E2awlT6sF6j3PZlc3xY4
fTtVm8PhLtqQ%3D%3D
Resolving storage.googleapis.com (storage.googleapis.com)... 172.217.212.128,
2607:f8b0:4001:c05::80
Connecting to storage.googleapis.com (storage.googleapis.com)|172.217.212.128
|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8408653 (8.0M) [application/zip]
Saving to: ‘train-v1.1.json.zip’

train-v1.1.json.zip 100%[===================>]   8.02M  --.-KB/s    in 0.1s

2019-08-25 16:08:26 (77.2 MB/s) - ‘train-v1.1.json.zip’ saved [8408653/840865
3]

FINISHED --2019-08-25 16:08:26--
Total wall clock time: 0.3s
Downloaded: 1 files, 8.0M in 0.1s (77.2 MB/s)
```

In [0]:
```
with zipfile.ZipFile("/content/train-v1.1.json.zip","r") as zip_ref:
    zip_ref.extractall()

with zipfile.ZipFile("/content/dev-v1.1.json.zip","r") as zip_ref:
    zip_ref.extractall()
```

In [0]:
```
#downloading weights and cofiguration file for the model
!wget https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip
```

```
--2019-08-25 05:17:25--  https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip
Resolving storage.googleapis.com (storage.googleapis.com)... 172.217.25.112, 2404:6800:4004:809::2010
Connecting to storage.googleapis.com (storage.googleapis.com)|172.217.25.112|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 407727028 (389M) [application/zip]
Saving to: ‘uncased_L-12_H-768_A-12.zip’

uncased_L-12_H-768_ 100%[===================>] 388.84M  78.1MB/s    in 5.0s

2019-08-25 05:17:31 (78.1 MB/s) - ‘uncased_L-12_H-768_A-12.zip’ saved [407727028/407727028]
```

Important Functions:

In [0]:
```
def read_squad_examples(input_file, is_training):
  """Read a SQuAD json file into a list of SquadExample."""
  with tf.gfile.Open(input_file, "r") as reader:
    input_data = json.load(reader)["data"]

  def is_whitespace(c):
    if c == " " or c == "\t" or c == "\r" or c == "\n" or ord(c) == 0x202F:
      return True
    return False

  examples = []
  for entry in input_data:
    for paragraph in entry["paragraphs"]:
      paragraph_text = paragraph["context"]
      doc_tokens = []
      char_to_word_offset = []
      prev_is_whitespace = True
      for c in paragraph_text:
        if is_whitespace(c):
          prev_is_whitespace = True
        else:
          if prev_is_whitespace:
            doc_tokens.append(c)
          else:
            doc_tokens[-1] += c
          prev_is_whitespace = False
        char_to_word_offset.append(len(doc_tokens) - 1)

      for qa in paragraph["qas"]:
        qas_id = qa["id"]
        question_text = qa["question"]
        start_position = None
        end_position = None
        orig_answer_text = None
        is_impossible = False
        if is_training:

          if False:
            is_impossible = qa["is_impossible"]
          if (len(qa["answers"]) != 1) and (not is_impossible):
            raise ValueError(
                "For training, each question should have exactly 1 answer.")
          if not is_impossible:
            answer = qa["answers"][0]
            orig_answer_text = answer["text"]
            answer_offset = answer["answer_start"]
            answer_length = len(orig_answer_text)
            start_position = char_to_word_offset[answer_offset]
            end_position = char_to_word_offset[answer_offset + answer_length -
                                               1]
            # Only add answers where the text can be exactly recovered from th
e
            # document. If this CAN'T happen it's likely due to weird Unicode
            # stuff so we will just skip the example.
            #
            # Note that this means for training mode, every example is NOT
```

```
            # guaranteed to be preserved.
            actual_text = " ".join(
                doc_tokens[start_position:(end_position + 1)])
            cleaned_answer_text = " ".join(
                tokenization.whitespace_tokenize(orig_answer_text))
            if actual_text.find(cleaned_answer_text) == -1:
              tf.logging.warning("Could not find answer: '%s' vs. '%s'",
                                 actual_text, cleaned_answer_text)
              continue
            else:
              start_position = -1
              end_position = -1
              orig_answer_text = ""

          example = run_squad.SquadExample(
              qas_id=qas_id,
              question_text=question_text,
              doc_tokens=doc_tokens,
              orig_answer_text=orig_answer_text,
              start_position=start_position,
              end_position=end_position,
              is_impossible=is_impossible)
          examples.append(example)

    return examples
```

In [0]:
```
def get_final_text(pred_text, orig_text, do_lower_case):
  """Project the tokenized prediction back to the original text."""

  def _strip_spaces(text):
    ns_chars = []
    ns_to_s_map = collections.OrderedDict()
    for (i, c) in enumerate(text):
      if c == " ":
        continue
      ns_to_s_map[len(ns_chars)] = i
      ns_chars.append(c)
    ns_text = "".join(ns_chars)
    return (ns_text, ns_to_s_map)

  # We first tokenize `orig_text`, strip whitespace from the result
  # and `pred_text`, and check if they are the same length. If they are
  # NOT the same length, the heuristic has failed. If they are the same
  # length, we assume the characters are one-to-one aligned.
  tokenizer = tokenization.BasicTokenizer(do_lower_case=do_lower_case)

  tok_text = " ".join(tokenizer.tokenize(orig_text))

  start_position = tok_text.find(pred_text)
  if start_position == -1:
    if True:
      tf.logging.info(
          "Unable to find text: '%s' in '%s'" % (pred_text, orig_text))
    return orig_text
  end_position = start_position + len(pred_text) - 1

  (orig_ns_text, orig_ns_to_s_map) = _strip_spaces(orig_text)
  (tok_ns_text, tok_ns_to_s_map) = _strip_spaces(tok_text)

  if len(orig_ns_text) != len(tok_ns_text):
    if True:
      tf.logging.info("Length not equal after stripping spaces: '%s' vs '%s'",
                      orig_ns_text, tok_ns_text)
    return orig_text

  # We then project the characters in `pred_text` back to `orig_text` using
  # the character-to-character alignment.
  tok_s_to_ns_map = {}
  for (i, tok_index) in six.iteritems(tok_ns_to_s_map):
    tok_s_to_ns_map[tok_index] = i

  orig_start_position = None
  if start_position in tok_s_to_ns_map:
    ns_start_position = tok_s_to_ns_map[start_position]
    if ns_start_position in orig_ns_to_s_map:
      orig_start_position = orig_ns_to_s_map[ns_start_position]

  if orig_start_position is None:
    if True:
      tf.logging.info("Couldn't map start position")
    return orig_text
```

```
  orig_end_position = None
  if end_position in tok_s_to_ns_map:
    ns_end_position = tok_s_to_ns_map[end_position]
    if ns_end_position in orig_ns_to_s_map:
      orig_end_position = orig_ns_to_s_map[ns_end_position]

  if orig_end_position is None:
    if True:
      tf.logging.info("Couldn't map end position")
    return orig_text

  output_text = orig_text[orig_start_position:(orig_end_position + 1)]
  return output_text
```

In [0]:
```
def write_predictions(all_examples, all_features, all_results, n_best_size,
                      max_answer_length, do_lower_case, output_prediction_file
,
                      output_nbest_file, output_null_log_odds_file):
  """Write final predictions to the json file and log-odds of null if neede
d."""
  tf.logging.info("Writing predictions to: %s" % (output_prediction_file))
  tf.logging.info("Writing nbest to: %s" % (output_nbest_file))

  example_index_to_features = collections.defaultdict(list)
  for feature in all_features:
    example_index_to_features[feature.example_index].append(feature)

  unique_id_to_result = {}
  for result in all_results:
    unique_id_to_result[result.unique_id] = result

  _PrelimPrediction = collections.namedtuple(  # pylint: disable=invalid-name
      "PrelimPrediction",
      ["feature_index", "start_index", "end_index", "start_logit", "end_logit"
])

  all_predictions = collections.OrderedDict()
  all_nbest_json = collections.OrderedDict()
  scores_diff_json = collections.OrderedDict()

  for (example_index, example) in enumerate(all_examples):
    features = example_index_to_features[example_index]

    prelim_predictions = []
    # keep track of the minimum score of null start+end of position 0
    score_null = 1000000  # large and positive
    min_null_feature_index = 0  # the paragraph slice with min mull score
    null_start_logit = 0  # the start logit at the slice with min null score
    null_end_logit = 0  # the end logit at the slice with min null score
    for (feature_index, feature) in enumerate(features):
      result = unique_id_to_result[feature.unique_id]
      start_indexes = run_squad._get_best_indexes(result.start_logits, n_best_
size)
      end_indexes = run_squad._get_best_indexes(result.end_logits, n_best_size
)
      # if we could have irrelevant answers, get the min score of irrelevant
      if False:
        feature_null_score = result.start_logits[0] + result.end_logits[0]
        if feature_null_score < score_null:
          score_null = feature_null_score
          min_null_feature_index = feature_index
          null_start_logit = result.start_logits[0]
          null_end_logit = result.end_logits[0]
      for start_index in start_indexes:
        for end_index in end_indexes:
          # We could hypothetically create invalid predictions, e.g., predict
          # that the start of the span is in the question. We throw out all
          # invalid predictions.
          if start_index >= len(feature.tokens):
            continue
```

```
      if end_index >= len(feature.tokens):
        continue
      if start_index not in feature.token_to_orig_map:
        continue
      if end_index not in feature.token_to_orig_map:
        continue
      if not feature.token_is_max_context.get(start_index, False):
        continue
      if end_index < start_index:
        continue
      length = end_index - start_index + 1
      if length > max_answer_length:
        continue
      prelim_predictions.append(
          _PrelimPrediction(
              feature_index=feature_index,
              start_index=start_index,
              end_index=end_index,
              start_logit=result.start_logits[start_index],
              end_logit=result.end_logits[end_index]))

  if False:
    prelim_predictions.append(
        _PrelimPrediction(
            feature_index=min_null_feature_index,
            start_index=0,
            end_index=0,
            start_logit=null_start_logit,
            end_logit=null_end_logit))
  prelim_predictions = sorted(
      prelim_predictions,
      key=lambda x: (x.start_logit + x.end_logit),
      reverse=True)

  _NbestPrediction = collections.namedtuple(  # pylint: disable=invalid-name
      "NbestPrediction", ["text", "start_logit", "end_logit"])

  seen_predictions = {}
  nbest = []
  for pred in prelim_predictions:
    if len(nbest) >= n_best_size:
      break
    feature = features[pred.feature_index]
    if pred.start_index > 0:  # this is a non-null prediction
      tok_tokens = feature.tokens[pred.start_index:(pred.end_index + 1)]
      orig_doc_start = feature.token_to_orig_map[pred.start_index]
      orig_doc_end = feature.token_to_orig_map[pred.end_index]
      orig_tokens = example.doc_tokens[orig_doc_start:(orig_doc_end + 1)]
      tok_text = " ".join(tok_tokens)

      # De-tokenize WordPieces that have been split off.
      tok_text = tok_text.replace(" ##", "")
      tok_text = tok_text.replace("##", "")

      # Clean whitespace
      tok_text = tok_text.strip()
      tok_text = " ".join(tok_text.split())
```

```
    orig_text = " ".join(orig_tokens)

    final_text = get_final_text(tok_text, orig_text, do_lower_case)
    if final_text in seen_predictions:
      continue

    seen_predictions[final_text] = True
  else:
    final_text = ""
    seen_predictions[final_text] = True

  nbest.append(
      _NbestPrediction(
          text=final_text,
          start_logit=pred.start_logit,
          end_logit=pred.end_logit))

# if we didn't inlude the empty option in the n-best, inlcude it
if False:
  if "" not in seen_predictions:
    nbest.append(
        _NbestPrediction(
            text="", start_logit=null_start_logit,
            end_logit=null_end_logit))
# In very rare edge cases we could have no valid predictions. So we
# just create a nonce prediction in this case to avoid failure.
if not nbest:
  nbest.append(
      _NbestPrediction(text="empty", start_logit=0.0, end_logit=0.0))

assert len(nbest) >= 1

total_scores = []
best_non_null_entry = None
for entry in nbest:
  total_scores.append(entry.start_logit + entry.end_logit)
  if not best_non_null_entry:
    if entry.text:
      best_non_null_entry = entry

probs = run_squad._compute_softmax(total_scores)

nbest_json = []
for (i, entry) in enumerate(nbest):
  output = collections.OrderedDict()
  output["text"] = entry.text
  output["probability"] = probs[i]
  output["start_logit"] = entry.start_logit
  output["end_logit"] = entry.end_logit
  nbest_json.append(output)

assert len(nbest_json) >= 1

if not False:
  all_predictions[example.qas_id] = nbest_json[0]["text"]
else:
  # predict "" iff the null score - the score of best non-null > threshold
```

```
      score_diff = score_null - best_non_null_entry.start_logit - (
          best_non_null_entry.end_logit)
      scores_diff_json[example.qas_id] = score_diff
      if score_diff > 0.0:
        all_predictions[example.qas_id] = ""
      else:
        all_predictions[example.qas_id] = best_non_null_entry.text

    all_nbest_json[example.qas_id] = nbest_json

  with tf.gfile.GFile(output_prediction_file, "w") as writer:
    writer.write(json.dumps(all_predictions, indent=4) + "\n")

  with tf.gfile.GFile(output_nbest_file, "w") as writer:
    writer.write(json.dumps(all_nbest_json, indent=4) + "\n")

  if False:
    with tf.gfile.GFile(output_null_log_odds_file, "w") as writer:
      writer.write(json.dumps(scores_diff_json, indent=4) + "\n")
```

In [0]:
```
tf.logging.set_verbosity(tf.logging.INFO)
```

In [15]:
```
train_examples = read_squad_examples("/content/train-v1.1.json",True)
print("Total train examples are ",len(train_examples))
```

```
Total train examples are  87599
```

Model Parameters

In [0]:
```
# Model Hyper Parameters
TRAIN_BATCH_SIZE = 16
LEARNING_RATE = 3e-5
NUM_TRAIN_EPOCHS = 2.0
WARMUP_PROPORTION = 0.1
MAX_SEQ_LENGTH = 256
EVAL_BATCH_SIZE = 8

tpu_cluster_resolver = None
SAVE_CHECKPOINTS_STEPS = 1000
ITERATIONS_PER_LOOP = 1000
NUM_TPU_CORES = 8
```

In [18]:
```
tokenizer = tokenization.FullTokenizer(vocab_file=VOCAB_FILE, do_lower_case=DO
_LOWER_CASE)
```

```
W0825 16:09:20.985759 140319166060416 deprecation_wrapper.py:119] From bert_r
epo/tokenization.py:125: The name tf.gfile.GFile is deprecated. Please use t
f.io.gfile.GFile instead.
```

In [0]:
```
num_train_steps = int(
    len(train_examples) / TRAIN_BATCH_SIZE * NUM_TRAIN_EPOCHS)
num_warmup_steps = int(num_train_steps * WARMUP_PROPORTION)
```

In [0]:
```
# Setup TPU related config
tpu_cluster_resolver = tf.contrib.cluster_resolver.TPUClusterResolver(TPU_ADDR
ESS)
NUM_TPU_CORES = 8
```

In [0]:
```
model_fn = run_squad.model_fn_builder(
    bert_config=modeling.BertConfig.from_json_file(CONFIG_FILE),
    init_checkpoint=INIT_CHECKPOINT,
    learning_rate=LEARNING_RATE,
    num_train_steps=num_train_steps,
    num_warmup_steps=num_warmup_steps,
    use_tpu=True, #If False training will fall on CPU or GPU, depending on wha
t is available
    use_one_hot_embeddings=True)
```

In [0]:
```
run_config = tf.contrib.tpu.RunConfig(
    cluster=tpu_cluster_resolver,
    model_dir=OUTPUT_DIR,
    save_checkpoints_steps=SAVE_CHECKPOINTS_STEPS,
    tpu_config=tf.contrib.tpu.TPUConfig(
        iterations_per_loop=ITERATIONS_PER_LOOP,
        num_shards=NUM_TPU_CORES,
        per_host_input_for_training=tf.contrib.tpu.InputPipelineConfig.PER_HOS
T_V2))
```

In [23]:
```
estimator = tf.contrib.tpu.TPUEstimator(
    use_tpu=True, #If False training will fall on CPU or GPU, depending on wha
t is available
    model_fn=model_fn,
    config=run_config,
    train_batch_size=TRAIN_BATCH_SIZE,
    predict_batch_size=EVAL_BATCH_SIZE,
    eval_batch_size=EVAL_BATCH_SIZE)
```

```
W0825 16:09:31.909059 140319166060416 estimator.py:1984] Estimator's model_fn
(<function model_fn_builder.<locals>.model_fn at 0x7f9e4c817510>) includes pa
rams argument, but params are not passed to Estimator.
I0825 16:09:31.916226 140319166060416 estimator.py:209] Using config: {'_mode
l_dir': 'gs://bert-on-squad/bert-checkpoints/models/SQUAD', '_tf_random_see
d': None, '_save_summary_steps': 100, '_save_checkpoints_steps': 1000, '_save
_checkpoints_secs': None, '_session_config': allow_soft_placement: true
cluster_def {
  job {
    name: "worker"
    tasks {
      key: 0
      value: "10.69.74.146:8470"
    }
  }
}
isolate_session_state: true
, '_keep_checkpoint_max': 5, '_keep_checkpoint_every_n_hours': 10000, '_log_s
tep_count_steps': None, '_train_distribute': None, '_device_fn': None, '_prot
ocol': None, '_eval_distribute': None, '_experimental_distribute': None, '_ex
perimental_max_worker_delay_secs': None, '_service': None, '_cluster_spec': <
tensorflow.python.training.server_lib.ClusterSpec object at 0x7f9e4ed5d048>,
'_task_type': 'worker', '_task_id': 0, '_global_id_in_cluster': 0, '_master':
'grpc://10.69.74.146:8470', '_evaluation_master': 'grpc://10.69.74.146:8470',
'_is_chief': True, '_num_ps_replicas': 0, '_num_worker_replicas': 1, '_tpu_co
nfig': TPUConfig(iterations_per_loop=1000, num_shards=8, num_cores_per_replic
a=None, per_host_input_for_training=3, tpu_job_name=None, initial_infeed_slee
p_secs=None, input_partition_dims=None, eval_training_input_configuration=2),
'_cluster': <tensorflow.python.distribute.cluster_resolver.tpu_cluster_resolv
er.TPUClusterResolver object at 0x7f9e4efc0390>}
I0825 16:09:31.917964 140319166060416 tpu_context.py:209] _TPUContext: eval_o
n_tpu True
```

In [24]:
```
print('Please wait...')
train_writer = run_squad.FeatureWriter(
        filename=os.path.join(OUTPUT_DIR, "train.tf_record"),
        is_training=True)


def append_feature(feature):
  train_features.append(feature)
  train_writer.process_feature(feature)
```

```
W0825 16:09:31.933366 140319166060416 deprecation_wrapper.py:119] From bert_r
epo/run_squad.py:1065: The name tf.python_io.TFRecordWriter is deprecated. Pl
ease use tf.io.TFRecordWriter instead.


Please wait...
```

In [25]:
```
train_features = []

run_squad.convert_examples_to_features(train_examples, tokenizer, MAX_SEQ_LENG
TH, 128, 64, True, output_fn=append_feature)

train_writer.close()
```

```
W0825 16:09:32.559788 140319166060416 deprecation_wrapper.py:119] From bert_r
epo/run_squad.py:431: The name tf.logging.info is deprecated. Please use tf.c
ompat.v1.logging.info instead.

I0825 16:09:32.561164 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.562616 140319166060416 run_squad.py:432] unique_id: 1000000000
I0825 16:09:32.563645 140319166060416 run_squad.py:433] example_index: 0
I0825 16:09:32.565001 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.565773 140319166060416 run_squad.py:436] tokens: [CLS] to whom
did the virgin mary allegedly appear in 1858 in lou ##rdes france ? [SEP] arc
hitectural ##ly , the school has a catholic character . atop the main buildin
g ' s gold dome is a golden statue of the virgin mary . immediately in front
of the main building and facing it , is a copper statue of christ with arms u
p ##rai ##sed with the legend " ve ##ni ##te ad me om ##nes " . next to the m
ain building is the basilica of the sacred heart . immediately behind the bas
ilica is the gr ##otto , a marian place of prayer and reflection . it is a re
plica of the gr ##otto at lou ##rdes , france where the virgin mary reputed #
#ly appeared to saint bern ##ade ##tte so ##ub ##iro ##us in 1858 . at the en
d of the main drive ( and in a direct line that connects through 3 statues an
d the gold dome ) , is a simple , modern stone statue of mary . [SEP]
I0825 16:09:32.567049 140319166060416 run_squad.py:438] token_to_orig_map: 1
7:0 18:0 19:0 20:1 21:2 22:3 23:4 24:5 25:6 26:6 27:7 28:8 29:9 30:10 31:10 3
2:10 33:11 34:12 35:13 36:14 37:15 38:16 39:17 40:18 41:19 42:20 43:20 44:21
45:22 46:23 47:24 48:25 49:26 50:27 51:28 52:29 53:30 54:30 55:31 56:32 57:33
58:34 59:35 60:36 61:37 62:38 63:39 64:39 65:39 66:40 67:41 68:42 69:43 70:43
71:43 72:43 73:44 74:45 75:46 76:46 77:46 78:46 79:47 80:48 81:49 82:50 83:51
84:52 85:53 86:54 87:55 88:56 89:57 90:58 91:58 92:59 93:60 94:61 95:62 96:63
97:64 98:65 99:65 100:65 101:66 102:67 103:68 104:69 105:70 106:71 107:72 10
8:72 109:73 110:74 111:75 112:76 113:77 114:78 115:79 116:79 117:80 118:81 11
9:81 120:81 121:82 122:83 123:84 124:85 125:86 126:87 127:87 128:88 129:89 13
0:90 131:91 132:91 133:91 134:92 135:92 136:92 137:92 138:93 139:94 140:94 14
1:95 142:96 143:97 144:98 145:99 146:100 147:101 148:102 149:102 150:103 151:
104 152:105 153:106 154:107 155:108 156:109 157:110 158:111 159:112 160:113 1
61:114 162:115 163:115 164:115 165:116 166:117 167:118 168:118 169:119 170:12
0 171:121 172:122 173:123 174:123
I0825 16:09:32.568131 140319166060416 run_squad.py:440] token_is_max_context:
17:True 18:True 19:True 20:True 21:True 22:True 23:True 24:True 25:True 26:Tr
ue 27:True 28:True 29:True 30:True 31:True 32:True 33:True 34:True 35:True 3
6:True 37:True 38:True 39:True 40:True 41:True 42:True 43:True 44:True 45:Tru
e 46:True 47:True 48:True 49:True 50:True 51:True 52:True 53:True 54:True 55:
True 56:True 57:True 58:True 59:True 60:True 61:True 62:True 63:True 64:True
65:True 66:True 67:True 68:True 69:True 70:True 71:True 72:True 73:True 74:Tr
ue 75:True 76:True 77:True 78:True 79:True 80:True 81:True 82:True 83:True 8
4:True 85:True 86:True 87:True 88:True 89:True 90:True 91:True 92:True 93:Tru
e 94:True 95:True 96:True 97:True 98:True 99:True 100:True 101:True 102:True
103:True 104:True 105:True 106:True 107:True 108:True 109:True 110:True 111:T
rue 112:True 113:True 114:True 115:True 116:True 117:True 118:True 119:True 1
20:True 121:True 122:True 123:True 124:True 125:True 126:True 127:True 128:Tr
ue 129:True 130:True 131:True 132:True 133:True 134:True 135:True 136:True 13
7:True 138:True 139:True 140:True 141:True 142:True 143:True 144:True 145:Tru
e 146:True 147:True 148:True 149:True 150:True 151:True 152:True 153:True 15
4:True 155:True 156:True 157:True 158:True 159:True 160:True 161:True 162:Tru
e 163:True 164:True 165:True 166:True 167:True 168:True 169:True 170:True 17
1:True 172:True 173:True 174:True
I0825 16:09:32.569059 140319166060416 run_squad.py:442] input_ids: 101 2000 3
183 2106 1996 6261 2984 9382 3711 1999 8517 1999 10223 26371 2605 1029 102 65
49 2135 1010 1996 2082 2038 1037 3234 2839 1012 10234 1996 2364 2311 1005 105
```

```
5 2751 8514 2003 1037 3585 6231 1997 1996 6261 2984 1012 3202 1999 2392 1997
1996 2364 2311 1998 5307 2009 1010 2003 1037 6967 6231 1997 4828 2007 2608 20
39 14995 6924 2007 1996 5722 1000 2310 3490 2618 4748 2033 18168 5267 1000 10
12 2279 2000 1996 2364 2311 2003 1996 13546 1997 1996 6730 2540 1012 3202 236
9 1996 13546 2003 1996 24665 23052 1010 1037 14042 2173 1997 7083 1998 9185 1
012 2009 2003 1037 15059 1997 1996 24665 23052 2012 10223 26371 1010 2605 207
3 1996 6261 2984 22353 2135 2596 2000 3002 16595 9648 4674 2061 12083 9711 22
71 1999 8517 1012 2012 1996 2203 1997 1996 2364 3298 1006 1998 1999 1037 3622
2240 2008 8539 2083 1017 11342 1998 1996 2751 8514 1007 1010 2003 1037 3722 1
010 2715 2962 6231 1997 2984 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.570190 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.571151 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.571782 140319166060416 run_squad.py:451] start_position: 130
I0825 16:09:32.572670 140319166060416 run_squad.py:452] end_position: 137
I0825 16:09:32.573306 140319166060416 run_squad.py:454] answer: saint bern ##
ade ##tte so ##ub ##iro ##us
I0825 16:09:32.579133 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.580312 140319166060416 run_squad.py:432] unique_id: 1000000001
I0825 16:09:32.581232 140319166060416 run_squad.py:433] example_index: 1
I0825 16:09:32.581898 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.582767 140319166060416 run_squad.py:436] tokens: [CLS] what is
in front of the notre dame main building ? [SEP] architectural ##ly , the sch
ool has a catholic character . atop the main building ' s gold dome is a gold
en statue of the virgin mary . immediately in front of the main building and
facing it , is a copper statue of christ with arms up ##rai ##sed with the le
gend " ve ##ni ##te ad me om ##nes " . next to the main building is the basil
ica of the sacred heart . immediately behind the basilica is the gr ##otto ,
a marian place of prayer and reflection . it is a replica of the gr ##otto at
lou ##rdes , france where the virgin mary reputed ##ly appeared to saint bern
##ade ##tte so ##ub ##iro ##us in 1858 . at the end of the main drive ( and i
n a direct line that connects through 3 statues and the gold dome ) , is a si
mple , modern stone statue of mary . [SEP]
I0825 16:09:32.583647 140319166060416 run_squad.py:438] token_to_orig_map: 1
3:0 14:0 15:0 16:1 17:2 18:3 19:4 20:5 21:6 22:6 23:7 24:8 25:9 26:10 27:10 2
8:10 29:11 30:12 31:13 32:14 33:15 34:16 35:17 36:18 37:19 38:20 39:20 40:21
41:22 42:23 43:24 44:25 45:26 46:27 47:28 48:29 49:30 50:30 51:31 52:32 53:33
54:34 55:35 56:36 57:37 58:38 59:39 60:39 61:39 62:40 63:41 64:42 65:43 66:43
67:43 68:43 69:44 70:45 71:46 72:46 73:46 74:46 75:47 76:48 77:49 78:50 79:51
80:52 81:53 82:54 83:55 84:56 85:57 86:58 87:58 88:59 89:60 90:61 91:62 92:63
93:64 94:65 95:65 96:65 97:66 98:67 99:68 100:69 101:70 102:71 103:72 104:72
105:73 106:74 107:75 108:76 109:77 110:78 111:79 112:79 113:80 114:81 115:81
```

```
116:81 117:82 118:83 119:84 120:85 121:86 122:87 123:87 124:88 125:89 126:90
127:91 128:91 129:91 130:92 131:92 132:92 133:92 134:93 135:94 136:94 137:95
138:96 139:97 140:98 141:99 142:100 143:101 144:102 145:102 146:103 147:104 1
48:105 149:106 150:107 151:108 152:109 153:110 154:111 155:112 156:113 157:11
4 158:115 159:115 160:115 161:116 162:117 163:118 164:118 165:119 166:120 16
7:121 168:122 169:123 170:123
I0825 16:09:32.584585 140319166060416 run_squad.py:440] token_is_max_context:
13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:True 22:Tr
ue 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:True 31:True 3
2:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 40:True 41:Tru
e 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:True 51:
True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:True 60:True
 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:True 70:T
rue 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:True 79:True 8
0:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 88:True 89:Tru
e 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:True 99:
True 100:True 101:True 102:True 103:True 104:True 105:True 106:True 107:True
 108:True 109:True 110:True 111:True 112:True 113:True 114:True 115:True 116:
True 117:True 118:True 119:True 120:True 121:True 122:True 123:True 124:True
 125:True 126:True 127:True 128:True 129:True 130:True 131:True 132:True 133:
True 134:True 135:True 136:True 137:True 138:True 139:True 140:True 141:True
 142:True 143:True 144:True 145:True 146:True 147:True 148:True 149:True 150:
True 151:True 152:True 153:True 154:True 155:True 156:True 157:True 158:True
 159:True 160:True 161:True 162:True 163:True 164:True 165:True 166:True 167:
True 168:True 169:True 170:True
I0825 16:09:32.585532 140319166060416 run_squad.py:442] input_ids: 101 2054 2
003 1999 2392 1997 1996 10289 8214 2364 2311 1029 102 6549 2135 1010 1996 208
2 2038 1037 3234 2839 1012 10234 1996 2364 2311 1005 1055 2751 8514 2003 1037
3585 6231 1997 1996 6261 2984 1012 3202 1999 2392 1997 1996 2364 2311 1998 53
07 2009 1010 2003 1037 6967 6231 1997 4828 2007 2608 2039 14995 6924 2007 199
6 5722 1000 2310 3490 2618 4748 2033 18168 5267 1000 1012 2279 2000 1996 2364
2311 2003 1996 13546 1997 1996 6730 2540 1012 3202 2369 1996 13546 2003 1996
 24665 23052 1010 1037 14042 2173 1997 7083 1998 9185 1012 2009 2003 1037 150
59 1997 1996 24665 23052 2012 10223 26371 1010 2605 2073 1996 6261 2984 22353
2135 2596 2000 3002 16595 9648 4674 2061 12083 9711 2271 1999 8517 1012 2012
 1996 2203 1997 1996 2364 3298 1006 1998 1999 1037 3622 2240 2008 8539 2083 1
017 11342 1998 1996 2751 8514 1007 1010 2003 1037 3722 1010 2715 2962 6231 19
97 2984 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.586230 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.587180 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.587784 140319166060416 run_squad.py:451] start_position: 52
```

```
I0825 16:09:32.588638 140319166060416 run_squad.py:452] end_position: 56
I0825 16:09:32.589303 140319166060416 run_squad.py:454] answer: a copper stat
ue of christ
I0825 16:09:32.595091 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.596207 140319166060416 run_squad.py:432] unique_id: 1000000002
I0825 16:09:32.597107 140319166060416 run_squad.py:433] example_index: 2
I0825 16:09:32.597732 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.598665 140319166060416 run_squad.py:436] tokens: [CLS] the bas
ilica of the sacred heart at notre dame is beside to which structure ? [SEP]
 architectural ##ly , the school has a catholic character . atop the main bui
lding ' s gold dome is a golden statue of the virgin mary . immediately in fr
ont of the main building and facing it , is a copper statue of christ with ar
ms up ##rai ##sed with the legend " ve ##ni ##te ad me om ##nes " . next to t
he main building is the basilica of the sacred heart . immediately behind the
basilica is the gr ##otto , a marian place of prayer and reflection . it is a
replica of the gr ##otto at lou ##rdes , france where the virgin mary reputed
##ly appeared to saint bern ##ade ##tte so ##ub ##iro ##us in 1858 . at the e
nd of the main drive ( and in a direct line that connects through 3 statues a
nd the gold dome ) , is a simple , modern stone statue of mary . [SEP]
I0825 16:09:32.599589 140319166060416 run_squad.py:438] token_to_orig_map: 1
7:0 18:0 19:0 20:1 21:2 22:3 23:4 24:5 25:6 26:6 27:7 28:8 29:9 30:10 31:10 3
2:10 33:11 34:12 35:13 36:14 37:15 38:16 39:17 40:18 41:19 42:20 43:20 44:21
 45:22 46:23 47:24 48:25 49:26 50:27 51:28 52:29 53:30 54:30 55:31 56:32 57:3
3 58:34 59:35 60:36 61:37 62:38 63:39 64:39 65:39 66:40 67:41 68:42 69:43 70:
43 71:43 72:43 73:44 74:45 75:46 76:46 77:46 78:46 79:47 80:48 81:49 82:50 8
3:51 84:52 85:53 86:54 87:55 88:56 89:57 90:58 91:58 92:59 93:60 94:61 95:62
 96:63 97:64 98:65 99:65 100:65 101:66 102:67 103:68 104:69 105:70 106:71 10
7:72 108:72 109:73 110:74 111:75 112:76 113:77 114:78 115:79 116:79 117:80 11
8:81 119:81 120:81 121:82 122:83 123:84 124:85 125:86 126:87 127:87 128:88 12
9:89 130:90 131:91 132:91 133:91 134:92 135:92 136:92 137:92 138:93 139:94 14
0:94 141:95 142:96 143:97 144:98 145:99 146:100 147:101 148:102 149:102 150:1
03 151:104 152:105 153:106 154:107 155:108 156:109 157:110 158:111 159:112 16
0:113 161:114 162:115 163:115 164:115 165:116 166:117 167:118 168:118 169:119
170:120 171:121 172:122 173:123 174:123
I0825 16:09:32.600312 140319166060416 run_squad.py:440] token_is_max_context:
17:True 18:True 19:True 20:True 21:True 22:True 23:True 24:True 25:True 26:Tr
ue 27:True 28:True 29:True 30:True 31:True 32:True 33:True 34:True 35:True 3
6:True 37:True 38:True 39:True 40:True 41:True 42:True 43:True 44:True 45:Tru
e 46:True 47:True 48:True 49:True 50:True 51:True 52:True 53:True 54:True 55:
True 56:True 57:True 58:True 59:True 60:True 61:True 62:True 63:True 64:True
 65:True 66:True 67:True 68:True 69:True 70:True 71:True 72:True 73:True 74:T
rue 75:True 76:True 77:True 78:True 79:True 80:True 81:True 82:True 83:True 8
4:True 85:True 86:True 87:True 88:True 89:True 90:True 91:True 92:True 93:Tru
e 94:True 95:True 96:True 97:True 98:True 99:True 100:True 101:True 102:True
 103:True 104:True 105:True 106:True 107:True 108:True 109:True 110:True 111:
True 112:True 113:True 114:True 115:True 116:True 117:True 118:True 119:True
 120:True 121:True 122:True 123:True 124:True 125:True 126:True 127:True 128:
True 129:True 130:True 131:True 132:True 133:True 134:True 135:True 136:True
 137:True 138:True 139:True 140:True 141:True 142:True 143:True 144:True 145:
True 146:True 147:True 148:True 149:True 150:True 151:True 152:True 153:True
 154:True 155:True 156:True 157:True 158:True 159:True 160:True 161:True 162:
True 163:True 164:True 165:True 166:True 167:True 168:True 169:True 170:True
 171:True 172:True 173:True 174:True
I0825 16:09:32.601671 140319166060416 run_squad.py:442] input_ids: 101 1996 1
3546 1997 1996 6730 2540 2012 10289 8214 2003 3875 2000 2029 3252 1029 102 65
49 2135 1010 1996 2082 2038 1037 3234 2839 1012 10234 1996 2364 2311 1005 105
5 2751 8514 2003 1037 3585 6231 1997 1996 6261 2984 1012 3202 1999 2392 1997
```

```
 1996 2364 2311 1998 5307 2009 1010 2003 1037 6967 6231 1997 4828 2007 2608 2
039 14995 6924 2007 1996 5722 1000 2310 3490 2618 4748 2033 18168 5267 1000 1
012 2279 2000 1996 2364 2311 2003 1996 13546 1997 1996 6730 2540 1012 3202 23
69 1996 13546 2003 1996 24665 23052 1010 1037 14042 2173 1997 7083 1998 9185
 1012 2009 2003 1037 15059 1997 1996 24665 23052 2012 10223 26371 1010 2605 2
073 1996 6261 2984 22353 2135 2596 2000 3002 16595 9648 4674 2061 12083 9711
 2271 1999 8517 1012 2012 1996 2203 1997 1996 2364 3298 1006 1998 1999 1037 3
622 2240 2008 8539 2083 1017 11342 1998 1996 2751 8514 1007 1010 2003 1037 37
22 1010 2715 2962 6231 1997 2984 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.602703 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.603398 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.604426 140319166060416 run_squad.py:451] start_position: 81
I0825 16:09:32.605193 140319166060416 run_squad.py:452] end_position: 83
I0825 16:09:32.605823 140319166060416 run_squad.py:454] answer: the main buil
ding
I0825 16:09:32.611356 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.612507 140319166060416 run_squad.py:432] unique_id: 1000000003
I0825 16:09:32.613187 140319166060416 run_squad.py:433] example_index: 3
I0825 16:09:32.614070 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.614719 140319166060416 run_squad.py:436] tokens: [CLS] what is
the gr ##otto at notre dame ? [SEP] architectural ##ly , the school has a cat
holic character . atop the main building ' s gold dome is a golden statue of
 the virgin mary . immediately in front of the main building and facing it ,
 is a copper statue of christ with arms up ##rai ##sed with the legend " ve #
#ni ##te ad me om ##nes " . next to the main building is the basilica of the
 sacred heart . immediately behind the basilica is the gr ##otto , a marian p
lace of prayer and reflection . it is a replica of the gr ##otto at lou ##rde
s , france where the virgin mary reputed ##ly appeared to saint bern ##ade ##
tte so ##ub ##iro ##us in 1858 . at the end of the main drive ( and in a dire
ct line that connects through 3 statues and the gold dome ) , is a simple , m
odern stone statue of mary . [SEP]
I0825 16:09:32.615617 140319166060416 run_squad.py:438] token_to_orig_map: 1
1:0 12:0 13:0 14:1 15:2 16:3 17:4 18:5 19:6 20:6 21:7 22:8 23:9 24:10 25:10 2
6:10 27:11 28:12 29:13 30:14 31:15 32:16 33:17 34:18 35:19 36:20 37:20 38:21
 39:22 40:23 41:24 42:25 43:26 44:27 45:28 46:29 47:30 48:30 49:31 50:32 51:3
3 52:34 53:35 54:36 55:37 56:38 57:39 58:39 59:39 60:40 61:41 62:42 63:43 64:
43 65:43 66:43 67:44 68:45 69:46 70:46 71:46 72:46 73:47 74:48 75:49 76:50 7
7:51 78:52 79:53 80:54 81:55 82:56 83:57 84:58 85:58 86:59 87:60 88:61 89:62
 90:63 91:64 92:65 93:65 94:65 95:66 96:67 97:68 98:69 99:70 100:71 101:72 10
2:72 103:73 104:74 105:75 106:76 107:77 108:78 109:79 110:79 111:80 112:81 11
3:81 114:81 115:82 116:83 117:84 118:85 119:86 120:87 121:87 122:88 123:89 12
```

```
4:90 125:91 126:91 127:91 128:92 129:92 130:92 131:92 132:93 133:94 134:94 13
5:95 136:96 137:97 138:98 139:99 140:100 141:101 142:102 143:102 144:103 145:
104 146:105 147:106 148:107 149:108 150:109 151:110 152:111 153:112 154:113 1
55:114 156:115 157:115 158:115 159:116 160:117 161:118 162:118 163:119 164:12
0 165:121 166:122 167:123 168:123
I0825 16:09:32.616492 140319166060416 run_squad.py:440] token_is_max_context:
11:True 12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:Tr
ue 21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:True 3
0:True 31:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 39:Tru
e 40:True 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:
True 50:True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:True
 59:True 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:T
rue 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:True 7
8:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 87:Tru
e 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:
True 98:True 99:True 100:True 101:True 102:True 103:True 104:True 105:True 10
6:True 107:True 108:True 109:True 110:True 111:True 112:True 113:True 114:Tru
e 115:True 116:True 117:True 118:True 119:True 120:True 121:True 122:True 12
3:True 124:True 125:True 126:True 127:True 128:True 129:True 130:True 131:Tru
e 132:True 133:True 134:True 135:True 136:True 137:True 138:True 139:True 14
0:True 141:True 142:True 143:True 144:True 145:True 146:True 147:True 148:Tru
e 149:True 150:True 151:True 152:True 153:True 154:True 155:True 156:True 15
7:True 158:True 159:True 160:True 161:True 162:True 163:True 164:True 165:Tru
e 166:True 167:True 168:True
I0825 16:09:32.617219 140319166060416 run_squad.py:442] input_ids: 101 2054 2
003 1996 24665 23052 2012 10289 8214 1029 102 6549 2135 1010 1996 2082 2038 1
037 3234 2839 1012 10234 1996 2364 2311 1005 1055 2751 8514 2003 1037 3585 62
31 1997 1996 6261 2984 1012 3202 1999 2392 1997 1996 2364 2311 1998 5307 2009
1010 2003 1037 6967 6231 1997 4828 2007 2608 2039 14995 6924 2007 1996 5722 1
000 2310 3490 2618 4748 2033 18168 5267 1000 1012 2279 2000 1996 2364 2311 20
03 1996 13546 1997 1996 6730 2540 1012 3202 2369 1996 13546 2003 1996 24665 2
3052 1010 1037 14042 2173 1997 7083 1998 9185 1012 2009 2003 1037 15059 1997
 1996 24665 23052 2012 10223 26371 1010 2605 2073 1996 6261 2984 22353 2135 2
596 2000 3002 16595 9648 4674 2061 12083 9711 2271 1999 8517 1012 2012 1996 2
203 1997 1996 2364 3298 1006 1998 1999 1037 3622 2240 2008 8539 2083 1017 113
42 1998 1996 2751 8514 1007 1010 2003 1037 3722 1010 2715 2962 6231 1997 2984
1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.618114 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.618770 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.619582 140319166060416 run_squad.py:451] start_position: 95
I0825 16:09:32.620209 140319166060416 run_squad.py:452] end_position: 101
```

```
I0825 16:09:32.621028 140319166060416 run_squad.py:454] answer: a marian plac
e of prayer and reflection
I0825 16:09:32.630326 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.631953 140319166060416 run_squad.py:432] unique_id: 1000000004
I0825 16:09:32.633031 140319166060416 run_squad.py:433] example_index: 4
I0825 16:09:32.633890 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.634968 140319166060416 run_squad.py:436] tokens: [CLS] what si
ts on top of the main building at notre dame ? [SEP] architectural ##ly , the
school has a catholic character . atop the main building ' s gold dome is a g
olden statue of the virgin mary . immediately in front of the main building a
nd facing it , is a copper statue of christ with arms up ##rai ##sed with the
legend " ve ##ni ##te ad me om ##nes " . next to the main building is the bas
ilica of the sacred heart . immediately behind the basilica is the gr ##otto
 , a marian place of prayer and reflection . it is a replica of the gr ##otto
at lou ##rdes , france where the virgin mary reputed ##ly appeared to saint b
ern ##ade ##tte so ##ub ##iro ##us in 1858 . at the end of the main drive ( a
nd in a direct line that connects through 3 statues and the gold dome ) , is
 a simple , modern stone statue of mary . [SEP]
I0825 16:09:32.635939 140319166060416 run_squad.py:438] token_to_orig_map: 1
4:0 15:0 16:0 17:1 18:2 19:3 20:4 21:5 22:6 23:6 24:7 25:8 26:9 27:10 28:10 2
9:10 30:11 31:12 32:13 33:14 34:15 35:16 36:17 37:18 38:19 39:20 40:20 41:21
 42:22 43:23 44:24 45:25 46:26 47:27 48:28 49:29 50:30 51:30 52:31 53:32 54:3
3 55:34 56:35 57:36 58:37 59:38 60:39 61:39 62:39 63:40 64:41 65:42 66:43 67:
43 68:43 69:43 70:44 71:45 72:46 73:46 74:46 75:46 76:47 77:48 78:49 79:50 8
0:51 81:52 82:53 83:54 84:55 85:56 86:57 87:58 88:58 89:59 90:60 91:61 92:62
 93:63 94:64 95:65 96:65 97:65 98:66 99:67 100:68 101:69 102:70 103:71 104:72
105:72 106:73 107:74 108:75 109:76 110:77 111:78 112:79 113:79 114:80 115:81
 116:81 117:81 118:82 119:83 120:84 121:85 122:86 123:87 124:87 125:88 126:89
127:90 128:91 129:91 130:91 131:92 132:92 133:92 134:92 135:93 136:94 137:94
 138:95 139:96 140:97 141:98 142:99 143:100 144:101 145:102 146:102 147:103 1
48:104 149:105 150:106 151:107 152:108 153:109 154:110 155:111 156:112 157:11
3 158:114 159:115 160:115 161:115 162:116 163:117 164:118 165:118 166:119 16
7:120 168:121 169:122 170:123 171:123
I0825 16:09:32.636708 140319166060416 run_squad.py:440] token_is_max_context:
14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:True 22:True 23:Tr
ue 24:True 25:True 26:True 27:True 28:True 29:True 30:True 31:True 32:True 3
3:True 34:True 35:True 36:True 37:True 38:True 39:True 40:True 41:True 42:Tru
e 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:True 51:True 52:
True 53:True 54:True 55:True 56:True 57:True 58:True 59:True 60:True 61:True
 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:True 70:True 71:T
rue 72:True 73:True 74:True 75:True 76:True 77:True 78:True 79:True 80:True 8
1:True 82:True 83:True 84:True 85:True 86:True 87:True 88:True 89:True 90:Tru
e 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:True 99:True 10
0:True 101:True 102:True 103:True 104:True 105:True 106:True 107:True 108:Tru
e 109:True 110:True 111:True 112:True 113:True 114:True 115:True 116:True 11
7:True 118:True 119:True 120:True 121:True 122:True 123:True 124:True 125:Tru
e 126:True 127:True 128:True 129:True 130:True 131:True 132:True 133:True 13
4:True 135:True 136:True 137:True 138:True 139:True 140:True 141:True 142:Tru
e 143:True 144:True 145:True 146:True 147:True 148:True 149:True 150:True 15
1:True 152:True 153:True 154:True 155:True 156:True 157:True 158:True 159:Tru
e 160:True 161:True 162:True 163:True 164:True 165:True 166:True 167:True 16
8:True 169:True 170:True 171:True
I0825 16:09:32.637784 140319166060416 run_squad.py:442] input_ids: 101 2054 7
719 2006 2327 1997 1996 2364 2311 2012 10289 8214 1029 102 6549 2135 1010 199
6 2082 2038 1037 3234 2839 1012 10234 1996 2364 2311 1005 1055 2751 8514 2003
1037 3585 6231 1997 1996 6261 2984 1012 3202 1999 2392 1997 1996 2364 2311 19
98 5307 2009 1010 2003 1037 6967 6231 1997 4828 2007 2608 2039 14995 6924 200
```

```
7 1996 5722 1000 2310 3490 2618 4748 2033 18168 5267 1000 1012 2279 2000 1996
2364 2311 2003 1996 13546 1997 1996 6730 2540 1012 3202 2369 1996 13546 2003
 1996 24665 23052 1010 1037 14042 2173 1997 7083 1998 9185 1012 2009 2003 103
7 15059 1997 1996 24665 23052 2012 10223 26371 1010 2605 2073 1996 6261 2984
 22353 2135 2596 2000 3002 16595 9648 4674 2061 12083 9711 2271 1999 8517 101
2 2012 1996 2203 1997 1996 2364 3298 1006 1998 1999 1037 3622 2240 2008 8539
 2083 1017 11342 1998 1996 2751 8514 1007 1010 2003 1037 3722 1010 2715 2962
 6231 1997 2984 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.638756 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.639518 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.640389 140319166060416 run_squad.py:451] start_position: 33
I0825 16:09:32.641024 140319166060416 run_squad.py:452] end_position: 39
I0825 16:09:32.641876 140319166060416 run_squad.py:454] answer: a golden stat
ue of the virgin mary
I0825 16:09:32.650233 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.651878 140319166060416 run_squad.py:432] unique_id: 1000000005
I0825 16:09:32.652780 140319166060416 run_squad.py:433] example_index: 5
I0825 16:09:32.653773 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.654778 140319166060416 run_squad.py:436] tokens: [CLS] when di
d the scholastic magazine of notre dame begin publishing ? [SEP] as at most o
ther universities , notre dame ' s students run a number of news media outlet
s . the nine student - run outlets include three newspapers , both a radio an
d television station , and several magazines and journals . begun as a one -
 page journal in september 1876 , the scholastic magazine is issued twice mon
thly and claims to be the oldest continuous collegiate publication in the uni
ted states . the other magazine , the jug ##gler , is released twice a year a
nd focuses on student literature and artwork . the dome yearbook is published
annually . the newspapers have varying publication interests , with the obser
ver published daily and mainly reporting university and other news , and staf
fed by students from both notre dame and saint mary ' s college . unlike scho
lastic and the dome , the observer is an independent publication and does not
have a faculty advisor or any editorial oversight from the university . in 19
87 , when some students believed that the observer began to show a conservati
ve bias , a liberal newspaper , common sense was published . likewise , in 20
03 , when other students believed that the paper showed a liberal bias , the
 conservative paper irish rover went into production . neither paper is publi
shed as often as the observer ; however , all three are distributed to all st
udents . finally , in spring 2008 an [SEP]
I0825 16:09:32.655658 140319166060416 run_squad.py:438] token_to_orig_map: 1
3:0 14:1 15:2 16:3 17:4 18:4 19:5 20:6 21:6 22:6 23:7 24:8 25:9 26:10 27:11 2
8:12 29:13 30:14 31:14 32:15 33:16 34:17 35:17 36:17 37:18 38:19 39:20 40:21
```

41:21 42:22 43:23 44:24 45:25 46:26 47:27 48:27 49:28 50:29 51:30 52:31 53:3
2 54:32 55:33 56:34 57:35 58:36 59:36 60:36 61:37 62:38 63:39 64:40 65:40 66:
41 67:42 68:43 69:44 70:45 71:46 72:47 73:48 74:49 75:50 76:51 77:52 78:53 7
9:54 80:55 81:56 82:57 83:58 84:59 85:60 86:60 87:61 88:62 89:63 90:63 91:64
92:65 93:65 94:65 95:66 96:67 97:68 98:69 99:70 100:71 101:72 102:73 103:74
104:75 105:76 106:77 107:77 108:78 109:79 110:80 111:81 112:82 113:83 114:83
115:84 116:85 117:86 118:87 119:88 120:89 121:89 122:90 123:91 124:92 125:93
126:94 127:95 128:96 129:97 130:98 131:99 132:100 133:101 134:101 135:102 13
6:103 137:104 138:105 139:106 140:107 141:108 142:109 143:110 144:111 145:112
146:112 147:112 148:113 149:113 150:114 151:115 152:116 153:117 154:118 155:1
18 156:119 157:120 158:121 159:122 160:123 161:124 162:125 163:126 164:127 16
5:128 166:129 167:130 168:131 169:132 170:133 171:134 172:135 173:136 174:137
175:138 176:138 177:139 178:140 179:140 180:141 181:142 182:143 183:144 184:1
45 185:146 186:147 187:148 188:149 189:150 190:151 191:152 192:153 193:153 19
4:154 195:155 196:156 197:156 198:157 199:158 200:159 201:160 202:160 203:161
204:161 205:162 206:163 207:163 208:164 209:165 210:166 211:167 212:168 213:1
69 214:170 215:171 216:172 217:173 218:174 219:174 220:175 221:176 222:177 22
3:178 224:179 225:180 226:181 227:182 228:182 229:183 230:184 231:185 232:186
233:187 234:188 235:189 236:190 237:191 238:191 239:192 240:192 241:193 242:1
94 243:195 244:196 245:197 246:198 247:199 248:199 249:200 250:200 251:201 25
2:202 253:203 254:204
I0825 16:09:32.656716 140319166060416 run_squad.py:440] token_is_max_context:
13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:True 22:Tr
ue 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:True 31:True 3
2:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 40:True 41:Tru
e 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:True 51:
True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:True 60:True
61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:True 70:T
rue 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:True 79:True 8
0:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 88:True 89:Tru
e 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:True 99:
True 100:True 101:True 102:True 103:True 104:True 105:True 106:True 107:True
108:True 109:True 110:True 111:True 112:True 113:True 114:True 115:True 116:
True 117:True 118:True 119:True 120:True 121:True 122:True 123:True 124:True
125:True 126:True 127:True 128:True 129:True 130:True 131:True 132:True 133:
True 134:True 135:True 136:True 137:True 138:True 139:True 140:True 141:True
142:True 143:True 144:True 145:True 146:True 147:True 148:True 149:True 150:
True 151:True 152:True 153:True 154:True 155:True 156:True 157:True 158:True
159:True 160:True 161:True 162:True 163:True 164:True 165:True 166:True 167:
True 168:True 169:True 170:True 171:True 172:True 173:True 174:True 175:True
176:True 177:True 178:True 179:True 180:True 181:True 182:True 183:True 184:
True 185:True 186:True 187:True 188:True 189:True 190:True 191:True 192:True
193:True 194:True 195:True 196:True 197:True 198:True 199:False 200:False 20
1:False 202:False 203:False 204:False 205:False 206:False 207:False 208:False
209:False 210:False 211:False 212:False 213:False 214:False 215:False 216:Fal
se 217:False 218:False 219:False 220:False 221:False 222:False 223:False 224:
False 225:False 226:False 227:False 228:False 229:False 230:False 231:False 2
32:False 233:False 234:False 235:False 236:False 237:False 238:False 239:Fals
e 240:False 241:False 242:False 243:False 244:False 245:False 246:False 247:F
alse 248:False 249:False 250:False 251:False 252:False 253:False 254:False
I0825 16:09:32.657735 140319166060416 run_squad.py:442] input_ids: 101 2043 2
106 1996 24105 2932 1997 10289 8214 4088 4640 1029 102 2004 2012 2087 2060 55
34 1010 10289 8214 1005 1055 2493 2448 1037 2193 1997 2739 2865 11730 1012 19
96 3157 3076 1011 2448 11730 2421 2093 6399 1010 2119 1037 2557 1998 2547 227
6 1010 1998 2195 7298 1998 9263 1012 5625 2004 1037 2028 1011 3931 3485 1999
2244 7326 1010 1996 24105 2932 2003 3843 3807 7058 1998 4447 2000 2022 1996
4587 7142 9234 4772 1999 1996 2142 2163 1012 1996 2060 2932 1010 1996 26536

```
 17420 1010 2003 2207 3807 1037 2095 1998 7679 2006 3076 3906 1998 8266 1012
 1996 8514 24803 2003 2405 6604 1012 1996 6399 2031 9671 4772 5426 1010 2007
 1996 9718 2405 3679 1998 3701 7316 2118 1998 2060 2739 1010 1998 21121 2011
 2493 2013 2119 10289 8214 1998 3002 2984 1005 1055 2267 1012 4406 24105 1998
1996 8514 1010 1996 9718 2003 2019 2981 4772 1998 2515 2025 2031 1037 4513 86
19 2030 2151 8368 15709 2013 1996 2118 1012 1999 3055 1010 2043 2070 2493 337
3 2008 1996 9718 2211 2000 2265 1037 4603 13827 1010 1037 4314 3780 1010 2691
3168 2001 2405 1012 10655 1010 1999 2494 1010 2043 2060 2493 3373 2008 1996 3
259 3662 1037 4314 13827 1010 1996 4603 3259 3493 13631 2253 2046 2537 1012 4
445 3259 2003 2405 2004 2411 2004 1996 9718 1025 2174 1010 2035 2093 2024 550
0 2000 2035 2493 1012 2633 1010 1999 3500 2263 2019 102
I0825 16:09:32.658510 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
I0825 16:09:32.659502 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
I0825 16:09:32.660385 140319166060416 run_squad.py:451] start_position: 63
I0825 16:09:32.661054 140319166060416 run_squad.py:452] end_position: 64
I0825 16:09:32.661916 140319166060416 run_squad.py:454] answer: september 187
6
I0825 16:09:32.664533 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.665292 140319166060416 run_squad.py:432] unique_id: 1000000006
I0825 16:09:32.666158 140319166060416 run_squad.py:433] example_index: 5
I0825 16:09:32.666749 140319166060416 run_squad.py:434] doc_span_index: 1
I0825 16:09:32.667675 140319166060416 run_squad.py:436] tokens: [CLS] when di
d the scholastic magazine of notre dame begin publishing ? [SEP] notre dame a
nd saint mary ' s college . unlike scholastic and the dome , the observer is
 an independent publication and does not have a faculty advisor or any editor
ial oversight from the university . in 1987 , when some students believed tha
t the observer began to show a conservative bias , a liberal newspaper , comm
on sense was published . likewise , in 2003 , when other students believed th
at the paper showed a liberal bias , the conservative paper irish rover went
 into production . neither paper is published as often as the observer ; howe
ver , all three are distributed to all students . finally , in spring 2008 an
undergraduate journal for political science research , beyond politics , made
its debut . [SEP]
I0825 16:09:32.668572 140319166060416 run_squad.py:438] token_to_orig_map: 1
3:108 14:109 15:110 16:111 17:112 18:112 19:112 20:113 21:113 22:114 23:115 2
4:116 25:117 26:118 27:118 28:119 29:120 30:121 31:122 32:123 33:124 34:125 3
5:126 36:127 37:128 38:129 39:130 40:131 41:132 42:133 43:134 44:135 45:136 4
6:137 47:138 48:138 49:139 50:140 51:140 52:141 53:142 54:143 55:144 56:145 5
7:146 58:147 59:148 60:149 61:150 62:151 63:152 64:153 65:153 66:154 67:155 6
8:156 69:156 70:157 71:158 72:159 73:160 74:160 75:161 76:161 77:162 78:163 7
9:163 80:164 81:165 82:166 83:167 84:168 85:169 86:170 87:171 88:172 89:173 9
0:174 91:174 92:175 93:176 94:177 95:178 96:179 97:180 98:181 99:182 100:182
 101:183 102:184 103:185 104:186 105:187 106:188 107:189 108:190 109:191 110:
```

```
191 111:192 112:192 113:193 114:194 115:195 116:196 117:197 118:198 119:199 1
20:199 121:200 122:200 123:201 124:202 125:203 126:204 127:205 128:206 129:20
7 130:208 131:209 132:210 133:210 134:211 135:212 136:212 137:213 138:214 13
9:215 140:215
I0825 16:09:32.669305 140319166060416 run_squad.py:440] token_is_max_context:
13:False 14:False 15:False 16:False 17:False 18:False 19:False 20:False 21:Fa
lse 22:False 23:False 24:False 25:False 26:False 27:False 28:False 29:False 3
0:False 31:False 32:False 33:False 34:False 35:False 36:False 37:False 38:Fal
se 39:False 40:False 41:False 42:False 43:False 44:False 45:False 46:False 4
7:False 48:False 49:False 50:False 51:False 52:False 53:False 54:False 55:Fal
se 56:False 57:False 58:False 59:False 60:False 61:False 62:False 63:False 6
4:False 65:False 66:False 67:False 68:False 69:False 70:False 71:True 72:True
73:True 74:True 75:True 76:True 77:True 78:True 79:True 80:True 81:True 82:Tr
ue 83:True 84:True 85:True 86:True 87:True 88:True 89:True 90:True 91:True 9
2:True 93:True 94:True 95:True 96:True 97:True 98:True 99:True 100:True 101:T
rue 102:True 103:True 104:True 105:True 106:True 107:True 108:True 109:True 1
10:True 111:True 112:True 113:True 114:True 115:True 116:True 117:True 118:Tr
ue 119:True 120:True 121:True 122:True 123:True 124:True 125:True 126:True 12
7:True 128:True 129:True 130:True 131:True 132:True 133:True 134:True 135:Tru
e 136:True 137:True 138:True 139:True 140:True
I0825 16:09:32.670263 140319166060416 run_squad.py:442] input_ids: 101 2043 2
106 1996 24105 2932 1997 10289 8214 4088 4640 1029 102 10289 8214 1998 3002 2
984 1005 1055 2267 1012 4406 24105 1998 1996 8514 1010 1996 9718 2003 2019 29
81 4772 1998 2515 2025 2031 1037 4513 8619 2030 2151 8368 15709 2013 1996 211
8 1012 1999 3055 1010 2043 2070 2493 3373 2008 1996 9718 2211 2000 2265 1037
 4603 13827 1010 1037 4314 3780 1010 2691 3168 2001 2405 1012 10655 1010 1999
2494 1010 2043 2060 2493 3373 2008 1996 3259 3662 1037 4314 13827 1010 1996 4
603 3259 3493 13631 2253 2046 2537 1012 4445 3259 2003 2405 2004 2411 2004 19
96 9718 1025 2174 1010 2035 2093 2024 5500 2000 2035 2493 1012 2633 1010 1999
3500 2263 2019 8324 3485 2005 2576 2671 2470 1010 3458 4331 1010 2081 2049 28
34 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
I0825 16:09:32.671016 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.671875 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.672505 140319166060416 run_squad.py:451] start_position: 0
I0825 16:09:32.673188 140319166060416 run_squad.py:452] end_position: 0
I0825 16:09:32.674167 140319166060416 run_squad.py:454] answer: [CLS]
I0825 16:09:32.682731 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.684496 140319166060416 run_squad.py:432] unique_id: 1000000007
I0825 16:09:32.685633 140319166060416 run_squad.py:433] example_index: 6
I0825 16:09:32.686831 140319166060416 run_squad.py:434] doc_span_index: 0
```

```
I0825 16:09:32.687821 140319166060416 run_squad.py:436] tokens: [CLS] how oft
en is notre dame ' s the jug ##gler published ? [SEP] as at most other univer
sities , notre dame ' s students run a number of news media outlets . the nin
e student - run outlets include three newspapers , both a radio and televisio
n station , and several magazines and journals . begun as a one - page journa
l in september 1876 , the scholastic magazine is issued twice monthly and cla
ims to be the oldest continuous collegiate publication in the united states .
the other magazine , the jug ##gler , is released twice a year and focuses on
student literature and artwork . the dome yearbook is published annually . th
e newspapers have varying publication interests , with the observer published
daily and mainly reporting university and other news , and staffed by student
s from both notre dame and saint mary ' s college . unlike scholastic and the
dome , the observer is an independent publication and does not have a faculty
advisor or any editorial oversight from the university . in 1987 , when some
 students believed that the observer began to show a conservative bias , a li
beral newspaper , common sense was published . likewise , in 2003 , when othe
r students believed that the paper showed a liberal bias , the conservative p
aper irish rover went into production . neither paper is published as often a
s the observer ; however , all three are distributed to all students . finall
y , in spring 2008 [SEP]
I0825 16:09:32.688614 140319166060416 run_squad.py:438] token_to_orig_map: 1
4:0 15:1 16:2 17:3 18:4 19:4 20:5 21:6 22:6 23:6 24:7 25:8 26:9 27:10 28:11 2
9:12 30:13 31:14 32:14 33:15 34:16 35:17 36:17 37:17 38:18 39:19 40:20 41:21
 42:21 43:22 44:23 45:24 46:25 47:26 48:27 49:27 50:28 51:29 52:30 53:31 54:3
2 55:32 56:33 57:34 58:35 59:36 60:36 61:36 62:37 63:38 64:39 65:40 66:40 67:
41 68:42 69:43 70:44 71:45 72:46 73:47 74:48 75:49 76:50 77:51 78:52 79:53 8
0:54 81:55 82:56 83:57 84:58 85:59 86:60 87:60 88:61 89:62 90:63 91:63 92:64
 93:65 94:65 95:65 96:66 97:67 98:68 99:69 100:70 101:71 102:72 103:73 104:74
105:75 106:76 107:77 108:77 109:78 110:79 111:80 112:81 113:82 114:83 115:83
 116:84 117:85 118:86 119:87 120:88 121:89 122:89 123:90 124:91 125:92 126:93
127:94 128:95 129:96 130:97 131:98 132:99 133:100 134:101 135:101 136:102 13
7:103 138:104 139:105 140:106 141:107 142:108 143:109 144:110 145:111 146:112
147:112 148:112 149:113 150:113 151:114 152:115 153:116 154:117 155:118 156:1
18 157:119 158:120 159:121 160:122 161:123 162:124 163:125 164:126 165:127 16
6:128 167:129 168:130 169:131 170:132 171:133 172:134 173:135 174:136 175:137
176:138 177:138 178:139 179:140 180:140 181:141 182:142 183:143 184:144 185:1
45 186:146 187:147 188:148 189:149 190:150 191:151 192:152 193:153 194:153 19
5:154 196:155 197:156 198:156 199:157 200:158 201:159 202:160 203:160 204:161
205:161 206:162 207:163 208:163 209:164 210:165 211:166 212:167 213:168 214:1
69 215:170 216:171 217:172 218:173 219:174 220:174 221:175 222:176 223:177 22
4:178 225:179 226:180 227:181 228:182 229:182 230:183 231:184 232:185 233:186
234:187 235:188 236:189 237:190 238:191 239:191 240:192 241:192 242:193 243:1
94 244:195 245:196 246:197 247:198 248:199 249:199 250:200 251:200 252:201 25
3:202 254:203
I0825 16:09:32.689622 140319166060416 run_squad.py:440] token_is_max_context:
14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:True 22:True 23:Tr
ue 24:True 25:True 26:True 27:True 28:True 29:True 30:True 31:True 32:True 3
3:True 34:True 35:True 36:True 37:True 38:True 39:True 40:True 41:True 42:Tru
e 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:True 51:True 52:
True 53:True 54:True 55:True 56:True 57:True 58:True 59:True 60:True 61:True
 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:True 70:True 71:T
rue 72:True 73:True 74:True 75:True 76:True 77:True 78:True 79:True 80:True 8
1:True 82:True 83:True 84:True 85:True 86:True 87:True 88:True 89:True 90:Tru
e 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:True 99:True 10
0:True 101:True 102:True 103:True 104:True 105:True 106:True 107:True 108:Tru
e 109:True 110:True 111:True 112:True 113:True 114:True 115:True 116:True 11
7:True 118:True 119:True 120:True 121:True 122:True 123:True 124:True 125:Tru
```

```
e 126:True 127:True 128:True 129:True 130:True 131:True 132:True 133:True 13
4:True 135:True 136:True 137:True 138:True 139:True 140:True 141:True 142:Tru
e 143:True 144:True 145:True 146:True 147:True 148:True 149:True 150:True 15
1:True 152:True 153:True 154:True 155:True 156:True 157:True 158:True 159:Tru
e 160:True 161:True 162:True 163:True 164:True 165:True 166:True 167:True 16
8:True 169:True 170:True 171:True 172:True 173:True 174:True 175:True 176:Tru
e 177:True 178:True 179:True 180:True 181:True 182:True 183:True 184:True 18
5:True 186:True 187:True 188:True 189:True 190:True 191:True 192:True 193:Tru
e 194:True 195:True 196:True 197:True 198:True 199:False 200:False 201:False
 202:False 203:False 204:False 205:False 206:False 207:False 208:False 209:Fa
lse 210:False 211:False 212:False 213:False 214:False 215:False 216:False 21
7:False 218:False 219:False 220:False 221:False 222:False 223:False 224:False
225:False 226:False 227:False 228:False 229:False 230:False 231:False 232:Fal
se 233:False 234:False 235:False 236:False 237:False 238:False 239:False 240:
False 241:False 242:False 243:False 244:False 245:False 246:False 247:False 2
48:False 249:False 250:False 251:False 252:False 253:False 254:False
I0825 16:09:32.690572 140319166060416 run_squad.py:442] input_ids: 101 2129 2
411 2003 10289 8214 1005 1055 1996 26536 17420 2405 1029 102 2004 2012 2087 2
060 5534 1010 10289 8214 1005 1055 2493 2448 1037 2193 1997 2739 2865 11730 1
012 1996 3157 3076 1011 2448 11730 2421 2093 6399 1010 2119 1037 2557 1998 25
47 2276 1010 1998 2195 7298 1998 9263 1012 5625 2004 1037 2028 1011 3931 3485
1999 2244 7326 1010 1996 24105 2932 2003 3843 3807 7058 1998 4447 2000 2022 1
996 4587 7142 9234 4772 1999 1996 2142 2163 1012 1996 2060 2932 1010 1996 265
36 17420 1010 2003 2207 3807 1037 2095 1998 7679 2006 3076 3906 1998 8266 101
2 1996 8514 24803 2003 2405 6604 1012 1996 6399 2031 9671 4772 5426 1010 2007
1996 9718 2405 3679 1998 3701 7316 2118 1998 2060 2739 1010 1998 21121 2011 2
493 2013 2119 10289 8214 1998 3002 2984 1005 1055 2267 1012 4406 24105 1998 1
996 8514 1010 1996 9718 2003 2019 2981 4772 1998 2515 2025 2031 1037 4513 861
9 2030 2151 8368 15709 2013 1996 2118 1012 1999 3055 1010 2043 2070 2493 3373
2008 1996 9718 2211 2000 2265 1037 4603 13827 1010 1037 4314 3780 1010 2691 3
168 2001 2405 1012 10655 1010 1999 2494 1010 2043 2060 2493 3373 2008 1996 32
59 3662 1037 4314 13827 1010 1996 4603 3259 3493 13631 2253 2046 2537 1012 44
45 3259 2003 2405 2004 2411 2004 1996 9718 1025 2174 1010 2035 2093 2024 5500
2000 2035 2493 1012 2633 1010 1999 3500 2263 102
I0825 16:09:32.691326 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
I0825 16:09:32.692248 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
I0825 16:09:32.692887 140319166060416 run_squad.py:451] start_position: 98
I0825 16:09:32.693682 140319166060416 run_squad.py:452] end_position: 98
I0825 16:09:32.694275 140319166060416 run_squad.py:454] answer: twice
I0825 16:09:32.696526 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.697507 140319166060416 run_squad.py:432] unique_id: 1000000008
I0825 16:09:32.698118 140319166060416 run_squad.py:433] example_index: 6
I0825 16:09:32.698932 140319166060416 run_squad.py:434] doc_span_index: 1
```

```
I0825 16:09:32.699573 140319166060416 run_squad.py:436] tokens: [CLS] how oft
en is notre dame ' s the jug ##gler published ? [SEP] notre dame and saint ma
ry ' s college . unlike scholastic and the dome , the observer is an independ
ent publication and does not have a faculty advisor or any editorial oversigh
t from the university . in 1987 , when some students believed that the observ
er began to show a conservative bias , a liberal newspaper , common sense was
published . likewise , in 2003 , when other students believed that the paper
 showed a liberal bias , the conservative paper irish rover went into product
ion . neither paper is published as often as the observer ; however , all thr
ee are distributed to all students . finally , in spring 2008 an undergraduat
e journal for political science research , beyond politics , made its debut .
[SEP]
I0825 16:09:32.700487 140319166060416 run_squad.py:438] token_to_orig_map: 1
4:108 15:109 16:110 17:111 18:112 19:112 20:112 21:113 22:113 23:114 24:115 2
5:116 26:117 27:118 28:118 29:119 30:120 31:121 32:122 33:123 34:124 35:125 3
6:126 37:127 38:128 39:129 40:130 41:131 42:132 43:133 44:134 45:135 46:136 4
7:137 48:138 49:138 50:139 51:140 52:140 53:141 54:142 55:143 56:144 57:145 5
8:146 59:147 60:148 61:149 62:150 63:151 64:152 65:153 66:153 67:154 68:155 6
9:156 70:156 71:157 72:158 73:159 74:160 75:160 76:161 77:161 78:162 79:163 8
0:163 81:164 82:165 83:166 84:167 85:168 86:169 87:170 88:171 89:172 90:173 9
1:174 92:174 93:175 94:176 95:177 96:178 97:179 98:180 99:181 100:182 101:182
102:183 103:184 104:185 105:186 106:187 107:188 108:189 109:190 110:191 111:1
91 112:192 113:192 114:193 115:194 116:195 117:196 118:197 119:198 120:199 12
1:199 122:200 123:200 124:201 125:202 126:203 127:204 128:205 129:206 130:207
131:208 132:209 133:210 134:210 135:211 136:212 137:212 138:213 139:214 140:2
15 141:215
I0825 16:09:32.701983 140319166060416 run_squad.py:440] token_is_max_context:
14:False 15:False 16:False 17:False 18:False 19:False 20:False 21:False 22:Fa
lse 23:False 24:False 25:False 26:False 27:False 28:False 29:False 30:False 3
1:False 32:False 33:False 34:False 35:False 36:False 37:False 38:False 39:Fal
se 40:False 41:False 42:False 43:False 44:False 45:False 46:False 47:False 4
8:False 49:False 50:False 51:False 52:False 53:False 54:False 55:False 56:Fal
se 57:False 58:False 59:False 60:False 61:False 62:False 63:False 64:False 6
5:False 66:False 67:False 68:False 69:False 70:False 71:True 72:True 73:True
 74:True 75:True 76:True 77:True 78:True 79:True 80:True 81:True 82:True 83:T
rue 84:True 85:True 86:True 87:True 88:True 89:True 90:True 91:True 92:True 9
3:True 94:True 95:True 96:True 97:True 98:True 99:True 100:True 101:True 102:
True 103:True 104:True 105:True 106:True 107:True 108:True 109:True 110:True
 111:True 112:True 113:True 114:True 115:True 116:True 117:True 118:True 119:
True 120:True 121:True 122:True 123:True 124:True 125:True 126:True 127:True
 128:True 129:True 130:True 131:True 132:True 133:True 134:True 135:True 136:
True 137:True 138:True 139:True 140:True 141:True
I0825 16:09:32.702727 140319166060416 run_squad.py:442] input_ids: 101 2129 2
411 2003 10289 8214 1005 1055 1996 26536 17420 2405 1029 102 10289 8214 1998
 3002 2984 1005 1055 2267 1012 4406 24105 1998 1996 8514 1010 1996 9718 2003
 2019 2981 4772 1998 2515 2025 2031 1037 4513 8619 2030 2151 8368 15709 2013
 1996 2118 1012 1999 3055 1010 2043 2070 2493 3373 2008 1996 9718 2211 2000 2
265 1037 4603 13827 1010 1037 4314 3780 1010 2691 3168 2001 2405 1012 10655 1
010 1999 2494 1010 2043 2060 2493 3373 2008 1996 3259 3662 1037 4314 13827 10
10 1996 4603 3259 3493 13631 2253 2046 2537 1012 4445 3259 2003 2405 2004 241
1 2004 1996 9718 1025 2174 1010 2035 2093 2024 5500 2000 2035 2493 1012 2633
 1010 1999 3500 2263 2019 8324 3485 2005 2576 2671 2470 1010 3458 4331 1010 2
081 2049 2834 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.703406 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
```

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.704361 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.705005 140319166060416 run_squad.py:451] start_position: 0
I0825 16:09:32.705590 140319166060416 run_squad.py:452] end_position: 0
I0825 16:09:32.706182 140319166060416 run_squad.py:454] answer: [CLS]
I0825 16:09:32.713954 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.715179 140319166060416 run_squad.py:432] unique_id: 1000000009
I0825 16:09:32.716102 140319166060416 run_squad.py:433] example_index: 7
I0825 16:09:32.716793 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.717685 140319166060416 run_squad.py:436] tokens: [CLS] what is
the daily student paper at notre dame called ? [SEP] as at most other univers
ities , notre dame ' s students run a number of news media outlets . the nine
student - run outlets include three newspapers , both a radio and television
 station , and several magazines and journals . begun as a one - page journal
in september 1876 , the scholastic magazine is issued twice monthly and claim
s to be the oldest continuous collegiate publication in the united states . t
he other magazine , the jug ##gler , is released twice a year and focuses on
 student literature and artwork . the dome yearbook is published annually . t
he newspapers have varying publication interests , with the observer publishe
d daily and mainly reporting university and other news , and staffed by stude
nts from both notre dame and saint mary ' s college . unlike scholastic and t
he dome , the observer is an independent publication and does not have a facu
lty advisor or any editorial oversight from the university . in 1987 , when s
ome students believed that the observer began to show a conservative bias , a
liberal newspaper , common sense was published . likewise , in 2003 , when ot
her students believed that the paper showed a liberal bias , the conservative
paper irish rover went into production . neither paper is published as often
 as the observer ; however , all three are distributed to all students . fina
lly , in spring 2008 an [SEP]
I0825 16:09:32.718493 140319166060416 run_squad.py:438] token_to_orig_map: 1
3:0 14:1 15:2 16:3 17:4 18:4 19:5 20:6 21:6 22:6 23:7 24:8 25:9 26:10 27:11 2
8:12 29:13 30:14 31:14 32:15 33:16 34:17 35:17 36:17 37:18 38:19 39:20 40:21
 41:21 42:22 43:23 44:24 45:25 46:26 47:27 48:27 49:28 50:29 51:30 52:31 53:3
2 54:32 55:33 56:34 57:35 58:36 59:36 60:36 61:37 62:38 63:39 64:40 65:40 66:
41 67:42 68:43 69:44 70:45 71:46 72:47 73:48 74:49 75:50 76:51 77:52 78:53 7
9:54 80:55 81:56 82:57 83:58 84:59 85:60 86:60 87:61 88:62 89:63 90:63 91:64
 92:65 93:65 94:65 95:66 96:67 97:68 98:69 99:70 100:71 101:72 102:73 103:74
 104:75 105:76 106:77 107:77 108:78 109:79 110:80 111:81 112:82 113:83 114:83
115:84 116:85 117:86 118:87 119:88 120:89 121:89 122:90 123:91 124:92 125:93
 126:94 127:95 128:96 129:97 130:98 131:99 132:100 133:101 134:101 135:102 13
6:103 137:104 138:105 139:106 140:107 141:108 142:109 143:110 144:111 145:112
146:112 147:112 148:113 149:113 150:114 151:115 152:116 153:117 154:118 155:1
18 156:119 157:120 158:121 159:122 160:123 161:124 162:125 163:126 164:127 16
5:128 166:129 167:130 168:131 169:132 170:133 171:134 172:135 173:136 174:137
```

```
175:138 176:138 177:139 178:140 179:140 180:141 181:142 182:143 183:144 184:1
45 185:146 186:147 187:148 188:149 189:150 190:151 191:152 192:153 193:153 19
4:154 195:155 196:156 197:156 198:157 199:158 200:159 201:160 202:160 203:161
204:161 205:162 206:163 207:163 208:164 209:165 210:166 211:167 212:168 213:1
69 214:170 215:171 216:172 217:173 218:174 219:174 220:175 221:176 222:177 22
3:178 224:179 225:180 226:181 227:182 228:182 229:183 230:184 231:185 232:186
233:187 234:188 235:189 236:190 237:191 238:191 239:192 240:192 241:193 242:1
94 243:195 244:196 245:197 246:198 247:199 248:199 249:200 250:200 251:201 25
2:202 253:203 254:204
I0825 16:09:32.719468 140319166060416 run_squad.py:440] token_is_max_context:
13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:True 22:Tr
ue 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:True 31:True 3
2:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 40:True 41:Tru
e 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:True 51:
True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:True 60:True
 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:True 70:T
rue 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:True 79:True 8
0:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 88:True 89:Tru
e 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:True 99:
True 100:True 101:True 102:True 103:True 104:True 105:True 106:True 107:True
 108:True 109:True 110:True 111:True 112:True 113:True 114:True 115:True 116:
True 117:True 118:True 119:True 120:True 121:True 122:True 123:True 124:True
 125:True 126:True 127:True 128:True 129:True 130:True 131:True 132:True 133:
True 134:True 135:True 136:True 137:True 138:True 139:True 140:True 141:True
 142:True 143:True 144:True 145:True 146:True 147:True 148:True 149:True 150:
True 151:True 152:True 153:True 154:True 155:True 156:True 157:True 158:True
 159:True 160:True 161:True 162:True 163:True 164:True 165:True 166:True 167:
True 168:True 169:True 170:True 171:True 172:True 173:True 174:True 175:True
 176:True 177:True 178:True 179:True 180:True 181:True 182:True 183:True 184:
True 185:True 186:True 187:True 188:True 189:True 190:True 191:True 192:True
 193:True 194:True 195:True 196:True 197:True 198:True 199:False 200:False 20
1:False 202:False 203:False 204:False 205:False 206:False 207:False 208:False
209:False 210:False 211:False 212:False 213:False 214:False 215:False 216:Fal
se 217:False 218:False 219:False 220:False 221:False 222:False 223:False 224:
False 225:False 226:False 227:False 228:False 229:False 230:False 231:False 2
32:False 233:False 234:False 235:False 236:False 237:False 238:False 239:Fals
e 240:False 241:False 242:False 243:False 244:False 245:False 246:False 247:F
alse 248:False 249:False 250:False 251:False 252:False 253:False 254:False
I0825 16:09:32.720269 140319166060416 run_squad.py:442] input_ids: 101 2054 2
003 1996 3679 3076 3259 2012 10289 8214 2170 1029 102 2004 2012 2087 2060 553
4 1010 10289 8214 1005 1055 2493 2448 1037 2193 1997 2739 2865 11730 1012 199
6 3157 3076 1011 2448 11730 2421 2093 6399 1010 2119 1037 2557 1998 2547 2276
1010 1998 2195 7298 1998 9263 1012 5625 2004 1037 2028 1011 3931 3485 1999 22
44 7326 1010 1996 24105 2932 2003 3843 3807 7058 1998 4447 2000 2022 1996 458
7 7142 9234 4772 1999 1996 2142 2163 1012 1996 2060 2932 1010 1996 26536 1742
0 1010 2003 2207 3807 1037 2095 1998 7679 2006 3076 3906 1998 8266 1012 1996
 8514 24803 2003 2405 6604 1012 1996 6399 2031 9671 4772 5426 1010 2007 1996
 9718 2405 3679 1998 3701 7316 2118 1998 2060 2739 1010 1998 21121 2011 2493
 2013 2119 10289 8214 1998 3002 2984 1005 1055 2267 1012 4406 24105 1998 1996
8514 1010 1996 9718 2003 2019 2981 4772 1998 2515 2025 2031 1037 4513 8619 20
30 2151 8368 15709 2013 1996 2118 1012 1999 3055 1010 2043 2070 2493 3373 200
8 1996 9718 2211 2000 2265 1037 4603 13827 1010 1037 4314 3780 1010 2691 3168
2001 2405 1012 10655 1010 1999 2494 1010 2043 2060 2493 3373 2008 1996 3259 3
662 1037 4314 13827 1010 1996 4603 3259 3493 13631 2253 2046 2537 1012 4445 3
259 2003 2405 2004 2411 2004 1996 9718 1025 2174 1010 2035 2093 2024 5500 200
0 2035 2493 1012 2633 1010 1999 3500 2263 2019 102
I0825 16:09:32.721209 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
```

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
I0825 16:09:32.721864 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
I0825 16:09:32.722712 140319166060416 run_squad.py:451] start_position: 123
I0825 16:09:32.723653 140319166060416 run_squad.py:452] end_position: 124
I0825 16:09:32.724304 140319166060416 run_squad.py:454] answer: the observer
I0825 16:09:32.726142 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.727061 140319166060416 run_squad.py:432] unique_id: 1000000010
I0825 16:09:32.727673 140319166060416 run_squad.py:433] example_index: 7
I0825 16:09:32.728520 140319166060416 run_squad.py:434] doc_span_index: 1
I0825 16:09:32.729214 140319166060416 run_squad.py:436] tokens: [CLS] what is
the daily student paper at notre dame called ? [SEP] notre dame and saint mar
y ' s college . unlike scholastic and the dome , the observer is an independe
nt publication and does not have a faculty advisor or any editorial oversight
from the university . in 1987 , when some students believed that the observer
began to show a conservative bias , a liberal newspaper , common sense was pu
blished . likewise , in 2003 , when other students believed that the paper sh
owed a liberal bias , the conservative paper irish rover went into production
. neither paper is published as often as the observer ; however , all three a
re distributed to all students . finally , in spring 2008 an undergraduate jo
urnal for political science research , beyond politics , made its debut . [SE
P]
I0825 16:09:32.730100 140319166060416 run_squad.py:438] token_to_orig_map: 1
3:108 14:109 15:110 16:111 17:112 18:112 19:112 20:113 21:113 22:114 23:115 2
4:116 25:117 26:118 27:118 28:119 29:120 30:121 31:122 32:123 33:124 34:125 3
5:126 36:127 37:128 38:129 39:130 40:131 41:132 42:133 43:134 44:135 45:136 4
6:137 47:138 48:138 49:139 50:140 51:140 52:141 53:142 54:143 55:144 56:145 5
7:146 58:147 59:148 60:149 61:150 62:151 63:152 64:153 65:153 66:154 67:155 6
8:156 69:156 70:157 71:158 72:159 73:160 74:160 75:161 76:161 77:162 78:163 7
9:163 80:164 81:165 82:166 83:167 84:168 85:169 86:170 87:171 88:172 89:173 9
0:174 91:174 92:175 93:176 94:177 95:178 96:179 97:180 98:181 99:182 100:182
 101:183 102:184 103:185 104:186 105:187 106:188 107:189 108:190 109:191 110:
191 111:192 112:192 113:193 114:194 115:195 116:196 117:197 118:198 119:199 1
20:199 121:200 122:200 123:201 124:202 125:203 126:204 127:205 128:206 129:20
7 130:208 131:209 132:210 133:210 134:211 135:212 136:212 137:213 138:214 13
9:215 140:215
I0825 16:09:32.730766 140319166060416 run_squad.py:440] token_is_max_context:
13:False 14:False 15:False 16:False 17:False 18:False 19:False 20:False 21:Fa
lse 22:False 23:False 24:False 25:False 26:False 27:False 28:False 29:False 3
0:False 31:False 32:False 33:False 34:False 35:False 36:False 37:False 38:Fal
se 39:False 40:False 41:False 42:False 43:False 44:False 45:False 46:False 4
7:False 48:False 49:False 50:False 51:False 52:False 53:False 54:False 55:Fal
se 56:False 57:False 58:False 59:False 60:False 61:False 62:False 63:False 6
4:False 65:False 66:False 67:False 68:False 69:False 70:False 71:True 72:True
73:True 74:True 75:True 76:True 77:True 78:True 79:True 80:True 81:True 82:Tr
```

```
ue 83:True 84:True 85:True 86:True 87:True 88:True 89:True 90:True 91:True 9
2:True 93:True 94:True 95:True 96:True 97:True 98:True 99:True 100:True 101:T
rue 102:True 103:True 104:True 105:True 106:True 107:True 108:True 109:True 1
10:True 111:True 112:True 113:True 114:True 115:True 116:True 117:True 118:Tr
ue 119:True 120:True 121:True 122:True 123:True 124:True 125:True 126:True 12
7:True 128:True 129:True 130:True 131:True 132:True 133:True 134:True 135:Tru
e 136:True 137:True 138:True 139:True 140:True
I0825 16:09:32.731711 140319166060416 run_squad.py:442] input_ids: 101 2054 2
003 1996 3679 3076 3259 2012 10289 8214 2170 1029 102 10289 8214 1998 3002 29
84 1005 1055 2267 1012 4406 24105 1998 1996 8514 1010 1996 9718 2003 2019 298
1 4772 1998 2515 2025 2031 1037 4513 8619 2030 2151 8368 15709 2013 1996 2118
1012 1999 3055 1010 2043 2070 2493 3373 2008 1996 9718 2211 2000 2265 1037 46
03 13827 1010 1037 4314 3780 1010 2691 3168 2001 2405 1012 10655 1010 1999 24
94 1010 2043 2060 2493 3373 2008 1996 3259 3662 1037 4314 13827 1010 1996 460
3 3259 3493 13631 2253 2046 2537 1012 4445 3259 2003 2405 2004 2411 2004 1996
9718 1025 2174 1010 2035 2093 2024 5500 2000 2035 2493 1012 2633 1010 1999 35
00 2263 2019 8324 3485 2005 2576 2671 2470 1010 3458 4331 1010 2081 2049 2834
1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0
I0825 16:09:32.732575 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.733252 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.734110 140319166060416 run_squad.py:451] start_position: 0
I0825 16:09:32.734893 140319166060416 run_squad.py:452] end_position: 0
I0825 16:09:32.735486 140319166060416 run_squad.py:454] answer: [CLS]
I0825 16:09:32.743881 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.745397 140319166060416 run_squad.py:432] unique_id: 1000000011
I0825 16:09:32.746536 140319166060416 run_squad.py:433] example_index: 8
I0825 16:09:32.747596 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.748450 140319166060416 run_squad.py:436] tokens: [CLS] how man
y student news papers are found at notre dame ? [SEP] as at most other univer
sities , notre dame ' s students run a number of news media outlets . the nin
e student - run outlets include three newspapers , both a radio and televisio
n station , and several magazines and journals . begun as a one - page journa
l in september 1876 , the scholastic magazine is issued twice monthly and cla
ims to be the oldest continuous collegiate publication in the united states .
the other magazine , the jug ##gler , is released twice a year and focuses on
student literature and artwork . the dome yearbook is published annually . th
e newspapers have varying publication interests , with the observer published
daily and mainly reporting university and other news , and staffed by student
s from both notre dame and saint mary ' s college . unlike scholastic and the
dome , the observer is an independent publication and does not have a faculty
```

```
advisor or any editorial oversight from the university . in 1987 , when some
 students believed that the observer began to show a conservative bias , a li
beral newspaper , common sense was published . likewise , in 2003 , when othe
r students believed that the paper showed a liberal bias , the conservative p
aper irish rover went into production . neither paper is published as often a
s the observer ; however , all three are distributed to all students . finall
y , in spring 2008 an [SEP]
I0825 16:09:32.749529 140319166060416 run_squad.py:438] token_to_orig_map: 1
3:0 14:1 15:2 16:3 17:4 18:4 19:5 20:6 21:6 22:6 23:7 24:8 25:9 26:10 27:11 2
8:12 29:13 30:14 31:14 32:15 33:16 34:17 35:17 36:17 37:18 38:19 39:20 40:21
 41:21 42:22 43:23 44:24 45:25 46:26 47:27 48:27 49:28 50:29 51:30 52:31 53:3
2 54:32 55:33 56:34 57:35 58:36 59:36 60:36 61:37 62:38 63:39 64:40 65:40 66:
41 67:42 68:43 69:44 70:45 71:46 72:47 73:48 74:49 75:50 76:51 77:52 78:53 7
9:54 80:55 81:56 82:57 83:58 84:59 85:60 86:60 87:61 88:62 89:63 90:63 91:64
 92:65 93:65 94:65 95:66 96:67 97:68 98:69 99:70 100:71 101:72 102:73 103:74
 104:75 105:76 106:77 107:77 108:78 109:79 110:80 111:81 112:82 113:83 114:83
115:84 116:85 117:86 118:87 119:88 120:89 121:89 122:90 123:91 124:92 125:93
 126:94 127:95 128:96 129:97 130:98 131:99 132:100 133:101 134:101 135:102 13
6:103 137:104 138:105 139:106 140:107 141:108 142:109 143:110 144:111 145:112
146:112 147:112 148:113 149:113 150:114 151:115 152:116 153:117 154:118 155:1
18 156:119 157:120 158:121 159:122 160:123 161:124 162:125 163:126 164:127 16
5:128 166:129 167:130 168:131 169:132 170:133 171:134 172:135 173:136 174:137
175:138 176:138 177:139 178:140 179:140 180:141 181:142 182:143 183:144 184:1
45 185:146 186:147 187:148 188:149 189:150 190:151 191:152 192:153 193:153 19
4:154 195:155 196:156 197:156 198:157 199:158 200:159 201:160 202:160 203:161
204:161 205:162 206:163 207:163 208:164 209:165 210:166 211:167 212:168 213:1
69 214:170 215:171 216:172 217:173 218:174 219:174 220:175 221:176 222:177 22
3:178 224:179 225:180 226:181 227:182 228:182 229:183 230:184 231:185 232:186
233:187 234:188 235:189 236:190 237:191 238:191 239:192 240:192 241:193 242:1
94 243:195 244:196 245:197 246:198 247:199 248:199 249:200 250:200 251:201 25
2:202 253:203 254:204
I0825 16:09:32.750642 140319166060416 run_squad.py:440] token_is_max_context:
13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:True 22:Tr
ue 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:True 31:True 3
2:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 40:True 41:Tru
e 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:True 51:
True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:True 60:True
 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:True 70:T
rue 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:True 79:True 8
0:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 88:True 89:Tru
e 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:True 99:
True 100:True 101:True 102:True 103:True 104:True 105:True 106:True 107:True
 108:True 109:True 110:True 111:True 112:True 113:True 114:True 115:True 116:
True 117:True 118:True 119:True 120:True 121:True 122:True 123:True 124:True
 125:True 126:True 127:True 128:True 129:True 130:True 131:True 132:True 133:
True 134:True 135:True 136:True 137:True 138:True 139:True 140:True 141:True
 142:True 143:True 144:True 145:True 146:True 147:True 148:True 149:True 150:
True 151:True 152:True 153:True 154:True 155:True 156:True 157:True 158:True
 159:True 160:True 161:True 162:True 163:True 164:True 165:True 166:True 167:
True 168:True 169:True 170:True 171:True 172:True 173:True 174:True 175:True
 176:True 177:True 178:True 179:True 180:True 181:True 182:True 183:True 184:
True 185:True 186:True 187:True 188:True 189:True 190:True 191:True 192:True
 193:True 194:True 195:True 196:True 197:True 198:True 199:False 200:False 20
1:False 202:False 203:False 204:False 205:False 206:False 207:False 208:False
209:False 210:False 211:False 212:False 213:False 214:False 215:False 216:Fal
se 217:False 218:False 219:False 220:False 221:False 222:False 223:False 224:
False 225:False 226:False 227:False 228:False 229:False 230:False 231:False 2
```

```
32:False 233:False 234:False 235:False 236:False 237:False 238:False 239:Fals
e 240:False 241:False 242:False 243:False 244:False 245:False 246:False 247:F
alse 248:False 249:False 250:False 251:False 252:False 253:False 254:False
I0825 16:09:32.751481 140319166060416 run_squad.py:442] input_ids: 101 2129 2
116 3076 2739 4981 2024 2179 2012 10289 8214 1029 102 2004 2012 2087 2060 553
4 1010 10289 8214 1005 1055 2493 2448 1037 2193 1997 2739 2865 11730 1012 199
6 3157 3076 1011 2448 11730 2421 2093 6399 1010 2119 1037 2557 1998 2547 2276
1010 1998 2195 7298 1998 9263 1012 5625 2004 1037 2028 1011 3931 3485 1999 22
44 7326 1010 1996 24105 2932 2003 3843 3807 7058 1998 4447 2000 2022 1996 458
7 7142 9234 4772 1999 1996 2142 2163 1012 1996 2060 2932 1010 1996 26536 1742
0 1010 2003 2207 3807 1037 2095 1998 7679 2006 3076 3906 1998 8266 1012 1996
 8514 24803 2003 2405 6604 1012 1996 6399 2031 9671 4772 5426 1010 2007 1996
 9718 2405 3679 1998 3701 7316 2118 1998 2060 2739 1010 1998 21121 2011 2493
 2013 2119 10289 8214 1998 3002 2984 1005 1055 2267 1012 4406 24105 1998 1996
8514 1010 1996 9718 2003 2019 2981 4772 1998 2515 2025 2031 1037 4513 8619 20
30 2151 8368 15709 2013 1996 2118 1012 1999 3055 1010 2043 2070 2493 3373 200
8 1996 9718 2211 2000 2265 1037 4603 13827 1010 1037 4314 3780 1010 2691 3168
2001 2405 1012 10655 1010 1999 2494 1010 2043 2060 2493 3373 2008 1996 3259 3
662 1037 4314 13827 1010 1996 4603 3259 3493 13631 2253 2046 2537 1012 4445 3
259 2003 2405 2004 2411 2004 1996 9718 1025 2174 1010 2035 2093 2024 5500 200
0 2035 2493 1012 2633 1010 1999 3500 2263 2019 102
I0825 16:09:32.752443 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
I0825 16:09:32.753219 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
I0825 16:09:32.754172 140319166060416 run_squad.py:451] start_position: 39
I0825 16:09:32.754981 140319166060416 run_squad.py:452] end_position: 39
I0825 16:09:32.755612 140319166060416 run_squad.py:454] answer: three
I0825 16:09:32.757464 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.758374 140319166060416 run_squad.py:432] unique_id: 1000000012
I0825 16:09:32.759002 140319166060416 run_squad.py:433] example_index: 8
I0825 16:09:32.759865 140319166060416 run_squad.py:434] doc_span_index: 1
I0825 16:09:32.760721 140319166060416 run_squad.py:436] tokens: [CLS] how man
y student news papers are found at notre dame ? [SEP] notre dame and saint ma
ry ' s college . unlike scholastic and the dome , the observer is an independ
ent publication and does not have a faculty advisor or any editorial oversigh
t from the university . in 1987 , when some students believed that the observ
er began to show a conservative bias , a liberal newspaper , common sense was
published . likewise , in 2003 , when other students believed that the paper
 showed a liberal bias , the conservative paper irish rover went into product
ion . neither paper is published as often as the observer ; however , all thr
ee are distributed to all students . finally , in spring 2008 an undergraduat
e journal for political science research , beyond politics , made its debut .
[SEP]
I0825 16:09:32.761457 140319166060416 run_squad.py:438] token_to_orig_map: 1
```

```
3:108 14:109 15:110 16:111 17:112 18:112 19:112 20:113 21:113 22:114 23:115 2
4:116 25:117 26:118 27:118 28:119 29:120 30:121 31:122 32:123 33:124 34:125 3
5:126 36:127 37:128 38:129 39:130 40:131 41:132 42:133 43:134 44:135 45:136 4
6:137 47:138 48:138 49:139 50:140 51:140 52:141 53:142 54:143 55:144 56:145 5
7:146 58:147 59:148 60:149 61:150 62:151 63:152 64:153 65:153 66:154 67:155 6
8:156 69:156 70:157 71:158 72:159 73:160 74:160 75:161 76:161 77:162 78:163 7
9:163 80:164 81:165 82:166 83:167 84:168 85:169 86:170 87:171 88:172 89:173 9
0:174 91:174 92:175 93:176 94:177 95:178 96:179 97:180 98:181 99:182 100:182
 101:183 102:184 103:185 104:186 105:187 106:188 107:189 108:190 109:191 110:
191 111:192 112:192 113:193 114:194 115:195 116:196 117:197 118:198 119:199 1
20:199 121:200 122:200 123:201 124:202 125:203 126:204 127:205 128:206 129:20
7 130:208 131:209 132:210 133:210 134:211 135:212 136:212 137:213 138:214 13
9:215 140:215
I0825 16:09:32.762403 140319166060416 run_squad.py:440] token_is_max_context:
13:False 14:False 15:False 16:False 17:False 18:False 19:False 20:False 21:Fa
lse 22:False 23:False 24:False 25:False 26:False 27:False 28:False 29:False 3
0:False 31:False 32:False 33:False 34:False 35:False 36:False 37:False 38:Fal
se 39:False 40:False 41:False 42:False 43:False 44:False 45:False 46:False 4
7:False 48:False 49:False 50:False 51:False 52:False 53:False 54:False 55:Fal
se 56:False 57:False 58:False 59:False 60:False 61:False 62:False 63:False 6
4:False 65:False 66:False 67:False 68:False 69:False 70:False 71:True 72:True
73:True 74:True 75:True 76:True 77:True 78:True 79:True 80:True 81:True 82:Tr
ue 83:True 84:True 85:True 86:True 87:True 88:True 89:True 90:True 91:True 9
2:True 93:True 94:True 95:True 96:True 97:True 98:True 99:True 100:True 101:T
rue 102:True 103:True 104:True 105:True 106:True 107:True 108:True 109:True 1
10:True 111:True 112:True 113:True 114:True 115:True 116:True 117:True 118:Tr
ue 119:True 120:True 121:True 122:True 123:True 124:True 125:True 126:True 12
7:True 128:True 129:True 130:True 131:True 132:True 133:True 134:True 135:Tru
e 136:True 137:True 138:True 139:True 140:True
I0825 16:09:32.763165 140319166060416 run_squad.py:442] input_ids: 101 2129 2
116 3076 2739 4981 2024 2179 2012 10289 8214 1029 102 10289 8214 1998 3002 29
84 1005 1055 2267 1012 4406 24105 1998 1996 8514 1010 1996 9718 2003 2019 298
1 4772 1998 2515 2025 2031 1037 4513 8619 2030 2151 8368 15709 2013 1996 2118
1012 1999 3055 1010 2043 2070 2493 3373 2008 1996 9718 2211 2000 2265 1037 46
03 13827 1010 1037 4314 3780 1010 2691 3168 2001 2405 1012 10655 1010 1999 24
94 1010 2043 2060 2493 3373 2008 1996 3259 3662 1037 4314 13827 1010 1996 460
3 3259 3493 13631 2253 2046 2537 1012 4445 3259 2003 2405 2004 2411 2004 1996
9718 1025 2174 1010 2035 2093 2024 5500 2000 2035 2493 1012 2633 1010 1999 35
00 2263 2019 8324 3485 2005 2576 2671 2470 1010 3458 4331 1010 2081 2049 2834
1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0
I0825 16:09:32.764236 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.765446 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.766068 140319166060416 run_squad.py:451] start_position: 0
I0825 16:09:32.767246 140319166060416 run_squad.py:452] end_position: 0
I0825 16:09:32.768428 140319166060416 run_squad.py:454] answer: [CLS]
I0825 16:09:32.788895 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.791840 140319166060416 run_squad.py:432] unique_id: 1000000013
I0825 16:09:32.795777 140319166060416 run_squad.py:433] example_index: 9
I0825 16:09:32.796664 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.797836 140319166060416 run_squad.py:436] tokens: [CLS] in what
year did the student paper common sense begin publication at notre dame ? [SE
P] as at most other universities , notre dame ' s students run a number of ne
ws media outlets . the nine student - run outlets include three newspapers ,
 both a radio and television station , and several magazines and journals . b
egun as a one - page journal in september 1876 , the scholastic magazine is i
ssued twice monthly and claims to be the oldest continuous collegiate publica
tion in the united states . the other magazine , the jug ##gler , is released
twice a year and focuses on student literature and artwork . the dome yearboo
k is published annually . the newspapers have varying publication interests ,
with the observer published daily and mainly reporting university and other n
ews , and staffed by students from both notre dame and saint mary ' s college
. unlike scholastic and the dome , the observer is an independent publication
and does not have a faculty advisor or any editorial oversight from the unive
rsity . in 1987 , when some students believed that the observer began to show
a conservative bias , a liberal newspaper , common sense was published . like
wise , in 2003 , when other students believed that the paper showed a liberal
bias , the conservative paper irish rover went into production . neither pape
r is published as often as the observer ; however , all three are distributed
to all students . finally , [SEP]
I0825 16:09:32.799245 140319166060416 run_squad.py:438] token_to_orig_map: 1
7:0 18:1 19:2 20:3 21:4 22:4 23:5 24:6 25:6 26:6 27:7 28:8 29:9 30:10 31:11 3
2:12 33:13 34:14 35:14 36:15 37:16 38:17 39:17 40:17 41:18 42:19 43:20 44:21
 45:21 46:22 47:23 48:24 49:25 50:26 51:27 52:27 53:28 54:29 55:30 56:31 57:3
2 58:32 59:33 60:34 61:35 62:36 63:36 64:36 65:37 66:38 67:39 68:40 69:40 70:
41 71:42 72:43 73:44 74:45 75:46 76:47 77:48 78:49 79:50 80:51 81:52 82:53 8
3:54 84:55 85:56 86:57 87:58 88:59 89:60 90:60 91:61 92:62 93:63 94:63 95:64
 96:65 97:65 98:65 99:66 100:67 101:68 102:69 103:70 104:71 105:72 106:73 10
7:74 108:75 109:76 110:77 111:77 112:78 113:79 114:80 115:81 116:82 117:83 11
8:83 119:84 120:85 121:86 122:87 123:88 124:89 125:89 126:90 127:91 128:92 12
9:93 130:94 131:95 132:96 133:97 134:98 135:99 136:100 137:101 138:101 139:10
2 140:103 141:104 142:105 143:106 144:107 145:108 146:109 147:110 148:111 14
9:112 150:112 151:112 152:113 153:113 154:114 155:115 156:116 157:117 158:118
159:118 160:119 161:120 162:121 163:122 164:123 165:124 166:125 167:126 168:1
27 169:128 170:129 171:130 172:131 173:132 174:133 175:134 176:135 177:136 17
8:137 179:138 180:138 181:139 182:140 183:140 184:141 185:142 186:143 187:144
188:145 189:146 190:147 191:148 192:149 193:150 194:151 195:152 196:153 197:1
53 198:154 199:155 200:156 201:156 202:157 203:158 204:159 205:160 206:160 20
7:161 208:161 209:162 210:163 211:163 212:164 213:165 214:166 215:167 216:168
217:169 218:170 219:171 220:172 221:173 222:174 223:174 224:175 225:176 226:1
77 227:178 228:179 229:180 230:181 231:182 232:182 233:183 234:184 235:185 23
6:186 237:187 238:188 239:189 240:190 241:191 242:191 243:192 244:192 245:193
246:194 247:195 248:196 249:197 250:198 251:199 252:199 253:200 254:200
I0825 16:09:32.800301 140319166060416 run_squad.py:440] token_is_max_context:
17:True 18:True 19:True 20:True 21:True 22:True 23:True 24:True 25:True 26:Tr
ue 27:True 28:True 29:True 30:True 31:True 32:True 33:True 34:True 35:True 3
6:True 37:True 38:True 39:True 40:True 41:True 42:True 43:True 44:True 45:Tru
e 46:True 47:True 48:True 49:True 50:True 51:True 52:True 53:True 54:True 55:
```

```
True 56:True 57:True 58:True 59:True 60:True 61:True 62:True 63:True 64:True
 65:True 66:True 67:True 68:True 69:True 70:True 71:True 72:True 73:True 74:T
rue 75:True 76:True 77:True 78:True 79:True 80:True 81:True 82:True 83:True 8
4:True 85:True 86:True 87:True 88:True 89:True 90:True 91:True 92:True 93:Tru
e 94:True 95:True 96:True 97:True 98:True 99:True 100:True 101:True 102:True
 103:True 104:True 105:True 106:True 107:True 108:True 109:True 110:True 111:
True 112:True 113:True 114:True 115:True 116:True 117:True 118:True 119:True
 120:True 121:True 122:True 123:True 124:True 125:True 126:True 127:True 128:
True 129:True 130:True 131:True 132:True 133:True 134:True 135:True 136:True
 137:True 138:True 139:True 140:True 141:True 142:True 143:True 144:True 145:
True 146:True 147:True 148:True 149:True 150:True 151:True 152:True 153:True
 154:True 155:True 156:True 157:True 158:True 159:True 160:True 161:True 162:
True 163:True 164:True 165:True 166:True 167:True 168:True 169:True 170:True
 171:True 172:True 173:True 174:True 175:True 176:True 177:True 178:True 179:
True 180:True 181:True 182:True 183:True 184:True 185:True 186:True 187:True
 188:True 189:True 190:True 191:True 192:True 193:True 194:True 195:True 196:
True 197:True 198:True 199:True 200:True 201:False 202:False 203:False 204:Fa
lse 205:False 206:False 207:False 208:False 209:False 210:False 211:False 21
2:False 213:False 214:False 215:False 216:False 217:False 218:False 219:False
220:False 221:False 222:False 223:False 224:False 225:False 226:False 227:Fal
se 228:False 229:False 230:False 231:False 232:False 233:False 234:False 235:
False 236:False 237:False 238:False 239:False 240:False 241:False 242:False 2
43:False 244:False 245:False 246:False 247:False 248:False 249:False 250:Fals
e 251:False 252:False 253:False 254:False
I0825 16:09:32.803601 140319166060416 run_squad.py:442] input_ids: 101 1999 2
054 2095 2106 1996 3076 3259 2691 3168 4088 4772 2012 10289 8214 1029 102 200
4 2012 2087 2060 5534 1010 10289 8214 1005 1055 2493 2448 1037 2193 1997 2739
2865 11730 1012 1996 3157 3076 1011 2448 11730 2421 2093 6399 1010 2119 1037
 2557 1998 2547 2276 1010 1998 2195 7298 1998 9263 1012 5625 2004 1037 2028 1
011 3931 3485 1999 2244 7326 1010 1996 24105 2932 2003 3843 3807 7058 1998 44
47 2000 2022 1996 4587 7142 9234 4772 1999 1996 2142 2163 1012 1996 2060 2932
1010 1996 26536 17420 1010 2003 2207 3807 1037 2095 1998 7679 2006 3076 3906
 1998 8266 1012 1996 8514 24803 2003 2405 6604 1012 1996 6399 2031 9671 4772
 5426 1010 2007 1996 9718 2405 3679 1998 3701 7316 2118 1998 2060 2739 1010 1
998 21121 2011 2493 2013 2119 10289 8214 1998 3002 2984 1005 1055 2267 1012 4
406 24105 1998 1996 8514 1010 1996 9718 2003 2019 2981 4772 1998 2515 2025 20
31 1037 4513 8619 2030 2151 8368 15709 2013 1996 2118 1012 1999 3055 1010 204
3 2070 2493 3373 2008 1996 9718 2211 2000 2265 1037 4603 13827 1010 1037 4314
3780 1010 2691 3168 2001 2405 1012 10655 1010 1999 2494 1010 2043 2060 2493 3
373 2008 1996 3259 3662 1037 4314 13827 1010 1996 4603 3259 3493 13631 2253 2
046 2537 1012 4445 3259 2003 2405 2004 2411 2004 1996 9718 1025 2174 1010 203
5 2093 2024 5500 2000 2035 2493 1012 2633 1010 102
I0825 16:09:32.805376 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
I0825 16:09:32.806914 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
I0825 16:09:32.809861 140319166060416 run_squad.py:451] start_position: 182
I0825 16:09:32.810606 140319166060416 run_squad.py:452] end_position: 182
I0825 16:09:32.811238 140319166060416 run_squad.py:454] answer: 1987
I0825 16:09:32.813205 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.814000 140319166060416 run_squad.py:432] unique_id: 1000000014
I0825 16:09:32.814635 140319166060416 run_squad.py:433] example_index: 9
I0825 16:09:32.815232 140319166060416 run_squad.py:434] doc_span_index: 1
I0825 16:09:32.815956 140319166060416 run_squad.py:436] tokens: [CLS] in what
year did the student paper common sense begin publication at notre dame ? [SE
P] notre dame and saint mary ' s college . unlike scholastic and the dome , t
he observer is an independent publication and does not have a faculty advisor
or any editorial oversight from the university . in 1987 , when some students
believed that the observer began to show a conservative bias , a liberal news
paper , common sense was published . likewise , in 2003 , when other students
believed that the paper showed a liberal bias , the conservative paper irish
 rover went into production . neither paper is published as often as the obse
rver ; however , all three are distributed to all students . finally , in spr
ing 2008 an undergraduate journal for political science research , beyond pol
itics , made its debut . [SEP]
I0825 16:09:32.816623 140319166060416 run_squad.py:438] token_to_orig_map: 1
7:108 18:109 19:110 20:111 21:112 22:112 23:112 24:113 25:113 26:114 27:115 2
8:116 29:117 30:118 31:118 32:119 33:120 34:121 35:122 36:123 37:124 38:125 3
9:126 40:127 41:128 42:129 43:130 44:131 45:132 46:133 47:134 48:135 49:136 5
0:137 51:138 52:138 53:139 54:140 55:140 56:141 57:142 58:143 59:144 60:145 6
1:146 62:147 63:148 64:149 65:150 66:151 67:152 68:153 69:153 70:154 71:155 7
2:156 73:156 74:157 75:158 76:159 77:160 78:160 79:161 80:161 81:162 82:163 8
3:163 84:164 85:165 86:166 87:167 88:168 89:169 90:170 91:171 92:172 93:173 9
4:174 95:174 96:175 97:176 98:177 99:178 100:179 101:180 102:181 103:182 104:
182 105:183 106:184 107:185 108:186 109:187 110:188 111:189 112:190 113:191 1
14:191 115:192 116:192 117:193 118:194 119:195 120:196 121:197 122:198 123:19
9 124:199 125:200 126:200 127:201 128:202 129:203 130:204 131:205 132:206 13
3:207 134:208 135:209 136:210 137:210 138:211 139:212 140:212 141:213 142:214
143:215 144:215
I0825 16:09:32.817566 140319166060416 run_squad.py:440] token_is_max_context:
17:False 18:False 19:False 20:False 21:False 22:False 23:False 24:False 25:Fa
lse 26:False 27:False 28:False 29:False 30:False 31:False 32:False 33:False 3
4:False 35:False 36:False 37:False 38:False 39:False 40:False 41:False 42:Fal
se 43:False 44:False 45:False 46:False 47:False 48:False 49:False 50:False 5
1:False 52:False 53:False 54:False 55:False 56:False 57:False 58:False 59:Fal
se 60:False 61:False 62:False 63:False 64:False 65:False 66:False 67:False 6
8:False 69:False 70:False 71:False 72:False 73:True 74:True 75:True 76:True 7
7:True 78:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True 86:Tru
e 87:True 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:
True 97:True 98:True 99:True 100:True 101:True 102:True 103:True 104:True 10
5:True 106:True 107:True 108:True 109:True 110:True 111:True 112:True 113:Tru
e 114:True 115:True 116:True 117:True 118:True 119:True 120:True 121:True 12
2:True 123:True 124:True 125:True 126:True 127:True 128:True 129:True 130:Tru
e 131:True 132:True 133:True 134:True 135:True 136:True 137:True 138:True 13
9:True 140:True 141:True 142:True 143:True 144:True
I0825 16:09:32.818270 140319166060416 run_squad.py:442] input_ids: 101 1999 2
054 2095 2106 1996 3076 3259 2691 3168 4088 4772 2012 10289 8214 1029 102 102
89 8214 1998 3002 2984 1005 1055 2267 1012 4406 24105 1998 1996 8514 1010 199
6 9718 2003 2019 2981 4772 1998 2515 2025 2031 1037 4513 8619 2030 2151 8368
 15709 2013 1996 2118 1012 1999 3055 1010 2043 2070 2493 3373 2008 1996 9718
 2211 2000 2265 1037 4603 13827 1010 1037 4314 3780 1010 2691 3168 2001 2405
 1012 10655 1010 1999 2494 1010 2043 2060 2493 3373 2008 1996 3259 3662 1037
```

```
 4314 13827 1010 1996 4603 3259 3493 13631 2253 2046 2537 1012 4445 3259 2003
2405 2004 2411 2004 1996 9718 1025 2174 1010 2035 2093 2024 5500 2000 2035 24
93 1012 2633 1010 1999 3500 2263 2019 8324 3485 2005 2576 2671 2470 1010 3458
4331 1010 2081 2049 2834 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.819248 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.819873 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.820752 140319166060416 run_squad.py:451] start_position: 54
I0825 16:09:32.821324 140319166060416 run_squad.py:452] end_position: 54
I0825 16:09:32.822116 140319166060416 run_squad.py:454] answer: 1987
I0825 16:09:32.827409 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.828511 140319166060416 run_squad.py:432] unique_id: 1000000015
I0825 16:09:32.829268 140319166060416 run_squad.py:433] example_index: 10
I0825 16:09:32.829890 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.830537 140319166060416 run_squad.py:436] tokens: [CLS] where i
s the headquarters of the congregation of the holy cross ? [SEP] the universi
ty is the major seat of the congregation of holy cross ( albeit not its offic
ial headquarters , which are in rome ) . its main seminary , more ##au semina
ry , is located on the campus across st . joseph lake from the main building
 . old college , the oldest building on campus and located near the shore of
 st . mary lake , houses undergraduate seminar ##ians . retired priests and b
rothers reside in fatima house ( a former retreat center ) , holy cross house
, as well as col ##umb ##a hall near the gr ##otto . the university through t
he more ##au seminary has ties to theologian frederick bu ##ech ##ner . while
not catholic , bu ##ech ##ner has praised writers from notre dame and more ##
au seminary created a bu ##ech ##ner prize for preaching . [SEP]
I0825 16:09:32.831253 140319166060416 run_squad.py:438] token_to_orig_map: 1
4:0 15:1 16:2 17:3 18:4 19:5 20:6 21:7 22:8 23:9 24:10 25:11 26:12 27:12 28:1
3 29:14 30:15 31:16 32:16 33:17 34:18 35:19 36:20 37:20 38:20 39:21 40:22 41:
23 42:23 43:24 44:24 45:25 46:25 47:26 48:27 49:28 50:29 51:30 52:31 53:32 5
4:32 55:33 56:34 57:35 58:36 59:37 60:38 61:38 62:39 63:40 64:40 65:41 66:42
 67:43 68:44 69:45 70:46 71:47 72:48 73:49 74:50 75:51 76:52 77:52 78:53 79:5
4 80:54 81:55 82:56 83:57 84:57 85:57 86:58 87:59 88:60 89:61 90:62 91:63 92:
64 93:65 94:66 95:66 96:67 97:68 98:69 99:69 100:69 101:70 102:71 103:72 104:
72 105:73 106:74 107:75 108:76 109:76 110:76 111:77 112:78 113:79 114:80 115:
80 116:80 117:81 118:82 119:83 120:84 121:85 122:85 123:86 124:87 125:88 126:
89 127:90 128:91 129:92 130:92 131:92 132:92 133:93 134:94 135:95 136:95 137:
96 138:96 139:96 140:97 141:98 142:99 143:100 144:101 145:102 146:103 147:104
148:104 149:105 150:106 151:107 152:108 153:108 154:108 155:109 156:110 157:1
11 158:111
I0825 16:09:32.831924 140319166060416 run_squad.py:440] token_is_max_context:
```

```
14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:True 22:True 23:Tr
ue 24:True 25:True 26:True 27:True 28:True 29:True 30:True 31:True 32:True 3
3:True 34:True 35:True 36:True 37:True 38:True 39:True 40:True 41:True 42:Tru
e 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:True 51:True 52:
True 53:True 54:True 55:True 56:True 57:True 58:True 59:True 60:True 61:True
 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:True 70:True 71:T
rue 72:True 73:True 74:True 75:True 76:True 77:True 78:True 79:True 80:True 8
1:True 82:True 83:True 84:True 85:True 86:True 87:True 88:True 89:True 90:Tru
e 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:True 99:True 10
0:True 101:True 102:True 103:True 104:True 105:True 106:True 107:True 108:Tru
e 109:True 110:True 111:True 112:True 113:True 114:True 115:True 116:True 11
7:True 118:True 119:True 120:True 121:True 122:True 123:True 124:True 125:Tru
e 126:True 127:True 128:True 129:True 130:True 131:True 132:True 133:True 13
4:True 135:True 136:True 137:True 138:True 139:True 140:True 141:True 142:Tru
e 143:True 144:True 145:True 146:True 147:True 148:True 149:True 150:True 15
1:True 152:True 153:True 154:True 155:True 156:True 157:True 158:True
I0825 16:09:32.832637 140319166060416 run_squad.py:442] input_ids: 101 2073 2
003 1996 4075 1997 1996 7769 1997 1996 4151 2892 1029 102 1996 2118 2003 1996
2350 2835 1997 1996 7769 1997 4151 2892 1006 12167 2025 2049 2880 4075 1010 2
029 2024 1999 4199 1007 1012 2049 2364 8705 1010 2062 4887 8705 1010 2003 228
4 2006 1996 3721 2408 2358 1012 3312 2697 2013 1996 2364 2311 1012 2214 2267
 1010 1996 4587 2311 2006 3721 1998 2284 2379 1996 5370 1997 2358 1012 2984 2
697 1010 3506 8324 18014 7066 1012 3394 8656 1998 3428 13960 1999 27596 2160
 1006 1037 2280 7822 2415 1007 1010 4151 2892 2160 1010 2004 2092 2004 8902 2
5438 2050 2534 2379 1996 24665 23052 1012 1996 2118 2083 1996 2062 4887 8705
 2038 7208 2000 17200 5406 20934 15937 3678 1012 2096 2025 3234 1010 20934 15
937 3678 2038 5868 4898 2013 10289 8214 1998 2062 4887 8705 2580 1037 20934 1
5937 3678 3396 2005 17979 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.833318 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.834311 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.834899 140319166060416 run_squad.py:451] start_position: 36
I0825 16:09:32.835547 140319166060416 run_squad.py:452] end_position: 36
I0825 16:09:32.836082 140319166060416 run_squad.py:454] answer: rome
I0825 16:09:32.841585 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.843089 140319166060416 run_squad.py:432] unique_id: 1000000016
I0825 16:09:32.844007 140319166060416 run_squad.py:433] example_index: 11
I0825 16:09:32.844674 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.846073 140319166060416 run_squad.py:436] tokens: [CLS] what is
the primary seminary of the congregation of the holy cross ? [SEP] the univer
sity is the major seat of the congregation of holy cross ( albeit not its off
icial headquarters , which are in rome ) . its main seminary , more ##au semi
```

```
nary , is located on the campus across st . joseph lake from the main buildin
g . old college , the oldest building on campus and located near the shore of
st . mary lake , houses undergraduate seminar ##ians . retired priests and br
others reside in fatima house ( a former retreat center ) , holy cross house
 , as well as col ##umb ##a hall near the gr ##otto . the university through
 the more ##au seminary has ties to theologian frederick bu ##ech ##ner . whi
le not catholic , bu ##ech ##ner has praised writers from notre dame and more
##au seminary created a bu ##ech ##ner prize for preaching . [SEP]
I0825 16:09:32.846788 140319166060416 run_squad.py:438] token_to_orig_map: 1
5:0 16:1 17:2 18:3 19:4 20:5 21:6 22:7 23:8 24:9 25:10 26:11 27:12 28:12 29:1
3 30:14 31:15 32:16 33:16 34:17 35:18 36:19 37:20 38:20 39:20 40:21 41:22 42:
23 43:23 44:24 45:24 46:25 47:25 48:26 49:27 50:28 51:29 52:30 53:31 54:32 5
5:32 56:33 57:34 58:35 59:36 60:37 61:38 62:38 63:39 64:40 65:40 66:41 67:42
 68:43 69:44 70:45 71:46 72:47 73:48 74:49 75:50 76:51 77:52 78:52 79:53 80:5
4 81:54 82:55 83:56 84:57 85:57 86:57 87:58 88:59 89:60 90:61 91:62 92:63 93:
64 94:65 95:66 96:66 97:67 98:68 99:69 100:69 101:69 102:70 103:71 104:72 10
5:72 106:73 107:74 108:75 109:76 110:76 111:76 112:77 113:78 114:79 115:80 11
6:80 117:80 118:81 119:82 120:83 121:84 122:85 123:85 124:86 125:87 126:88 12
7:89 128:90 129:91 130:92 131:92 132:92 133:92 134:93 135:94 136:95 137:95 13
8:96 139:96 140:96 141:97 142:98 143:99 144:100 145:101 146:102 147:103 148:1
04 149:104 150:105 151:106 152:107 153:108 154:108 155:108 156:109 157:110 15
8:111 159:111
I0825 16:09:32.847712 140319166060416 run_squad.py:440] token_is_max_context:
15:True 16:True 17:True 18:True 19:True 20:True 21:True 22:True 23:True 24:Tr
ue 25:True 26:True 27:True 28:True 29:True 30:True 31:True 32:True 33:True 3
4:True 35:True 36:True 37:True 38:True 39:True 40:True 41:True 42:True 43:Tru
e 44:True 45:True 46:True 47:True 48:True 49:True 50:True 51:True 52:True 53:
True 54:True 55:True 56:True 57:True 58:True 59:True 60:True 61:True 62:True
 63:True 64:True 65:True 66:True 67:True 68:True 69:True 70:True 71:True 72:T
rue 73:True 74:True 75:True 76:True 77:True 78:True 79:True 80:True 81:True 8
2:True 83:True 84:True 85:True 86:True 87:True 88:True 89:True 90:True 91:Tru
e 92:True 93:True 94:True 95:True 96:True 97:True 98:True 99:True 100:True 10
1:True 102:True 103:True 104:True 105:True 106:True 107:True 108:True 109:Tru
e 110:True 111:True 112:True 113:True 114:True 115:True 116:True 117:True 11
8:True 119:True 120:True 121:True 122:True 123:True 124:True 125:True 126:Tru
e 127:True 128:True 129:True 130:True 131:True 132:True 133:True 134:True 13
5:True 136:True 137:True 138:True 139:True 140:True 141:True 142:True 143:Tru
e 144:True 145:True 146:True 147:True 148:True 149:True 150:True 151:True 15
2:True 153:True 154:True 155:True 156:True 157:True 158:True 159:True
I0825 16:09:32.848484 140319166060416 run_squad.py:442] input_ids: 101 2054 2
003 1996 3078 8705 1997 1996 7769 1997 1996 4151 2892 1029 102 1996 2118 2003
1996 2350 2835 1997 1996 7769 1997 4151 2892 1006 12167 2025 2049 2880 4075 1
010 2029 2024 1999 4199 1007 1012 2049 2364 8705 1010 2062 4887 8705 1010 200
3 2284 2006 1996 3721 2408 2358 1012 3312 2697 2013 1996 2364 2311 1012 2214
 2267 1010 1996 4587 2311 2006 3721 1998 2284 2379 1996 5370 1997 2358 1012 2
984 2697 1010 3506 8324 18014 7066 1012 3394 8656 1998 3428 13960 1999 27596
 2160 1006 1037 2280 7822 2415 1007 1010 4151 2892 2160 1010 2004 2092 2004 8
902 25438 2050 2534 2379 1996 24665 23052 1012 1996 2118 2083 1996 2062 4887
 8705 2038 7208 2000 17200 5406 20934 15937 3678 1012 2096 2025 3234 1010 209
34 15937 3678 2038 5868 4898 2013 10289 8214 1998 2062 4887 8705 2580 1037 20
934 15937 3678 3396 2005 17979 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.849206 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.850232 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.850892 140319166060416 run_squad.py:451] start_position: 44
I0825 16:09:32.852055 140319166060416 run_squad.py:452] end_position: 46
I0825 16:09:32.852996 140319166060416 run_squad.py:454] answer: more ##au sem
inary
I0825 16:09:32.859001 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.860327 140319166060416 run_squad.py:432] unique_id: 1000000017
I0825 16:09:32.861329 140319166060416 run_squad.py:433] example_index: 12
I0825 16:09:32.862025 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.863269 140319166060416 run_squad.py:436] tokens: [CLS] what is
the oldest structure at notre dame ? [SEP] the university is the major seat o
f the congregation of holy cross ( albeit not its official headquarters , whi
ch are in rome ) . its main seminary , more ##au seminary , is located on the
campus across st . joseph lake from the main building . old college , the old
est building on campus and located near the shore of st . mary lake , houses
 undergraduate seminar ##ians . retired priests and brothers reside in fatima
house ( a former retreat center ) , holy cross house , as well as col ##umb #
#a hall near the gr ##otto . the university through the more ##au seminary ha
s ties to theologian frederick bu ##ech ##ner . while not catholic , bu ##ech
##ner has praised writers from notre dame and more ##au seminary created a bu
##ech ##ner prize for preaching . [SEP]
I0825 16:09:32.864354 140319166060416 run_squad.py:438] token_to_orig_map: 1
1:0 12:1 13:2 14:3 15:4 16:5 17:6 18:7 19:8 20:9 21:10 22:11 23:12 24:12 25:1
3 26:14 27:15 28:16 29:16 30:17 31:18 32:19 33:20 34:20 35:20 36:21 37:22 38:
23 39:23 40:24 41:24 42:25 43:25 44:26 45:27 46:28 47:29 48:30 49:31 50:32 5
1:32 52:33 53:34 54:35 55:36 56:37 57:38 58:38 59:39 60:40 61:40 62:41 63:42
 64:43 65:44 66:45 67:46 68:47 69:48 70:49 71:50 72:51 73:52 74:52 75:53 76:5
4 77:54 78:55 79:56 80:57 81:57 82:57 83:58 84:59 85:60 86:61 87:62 88:63 89:
64 90:65 91:66 92:66 93:67 94:68 95:69 96:69 97:69 98:70 99:71 100:72 101:72
 102:73 103:74 104:75 105:76 106:76 107:76 108:77 109:78 110:79 111:80 112:80
113:80 114:81 115:82 116:83 117:84 118:85 119:85 120:86 121:87 122:88 123:89
 124:90 125:91 126:92 127:92 128:92 129:92 130:93 131:94 132:95 133:95 134:96
135:96 136:96 137:97 138:98 139:99 140:100 141:101 142:102 143:103 144:104 14
5:104 146:105 147:106 148:107 149:108 150:108 151:108 152:109 153:110 154:111
155:111
I0825 16:09:32.865276 140319166060416 run_squad.py:440] token_is_max_context:
11:True 12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:Tr
ue 21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:True 3
0:True 31:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 39:Tru
e 40:True 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:
True 50:True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:True
 59:True 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:T
rue 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:True 7
8:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 87:Tru
e 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:
True 98:True 99:True 100:True 101:True 102:True 103:True 104:True 105:True 10
```

```
6:True 107:True 108:True 109:True 110:True 111:True 112:True 113:True 114:Tru
e 115:True 116:True 117:True 118:True 119:True 120:True 121:True 122:True 12
3:True 124:True 125:True 126:True 127:True 128:True 129:True 130:True 131:Tru
e 132:True 133:True 134:True 135:True 136:True 137:True 138:True 139:True 14
0:True 141:True 142:True 143:True 144:True 145:True 146:True 147:True 148:Tru
e 149:True 150:True 151:True 152:True 153:True 154:True 155:True
I0825 16:09:32.866091 140319166060416 run_squad.py:442] input_ids: 101 2054 2
003 1996 4587 3252 2012 10289 8214 1029 102 1996 2118 2003 1996 2350 2835 199
7 1996 7769 1997 4151 2892 1006 12167 2025 2049 2880 4075 1010 2029 2024 1999
4199 1007 1012 2049 2364 8705 1010 2062 4887 8705 1010 2003 2284 2006 1996 37
21 2408 2358 1012 3312 2697 2013 1996 2364 2311 1012 2214 2267 1010 1996 4587
2311 2006 3721 1998 2284 2379 1996 5370 1997 2358 1012 2984 2697 1010 3506 83
24 18014 7066 1012 3394 8656 1998 3428 13960 1999 27596 2160 1006 1037 2280 7
822 2415 1007 1010 4151 2892 2160 1010 2004 2092 2004 8902 25438 2050 2534 23
79 1996 24665 23052 1012 1996 2118 2083 1996 2062 4887 8705 2038 7208 2000 17
200 5406 20934 15937 3678 1012 2096 2025 3234 1010 20934 15937 3678 2038 5868
4898 2013 10289 8214 1998 2062 4887 8705 2580 1037 20934 15937 3678 3396 2005
17979 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.867258 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.868218 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.869050 140319166060416 run_squad.py:451] start_position: 59
I0825 16:09:32.869669 140319166060416 run_squad.py:452] end_position: 60
I0825 16:09:32.870445 140319166060416 run_squad.py:454] answer: old college
I0825 16:09:32.876367 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.877520 140319166060416 run_squad.py:432] unique_id: 1000000018
I0825 16:09:32.878204 140319166060416 run_squad.py:433] example_index: 13
I0825 16:09:32.879137 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.879998 140319166060416 run_squad.py:436] tokens: [CLS] what in
dividuals live at fatima house at notre dame ? [SEP] the university is the ma
jor seat of the congregation of holy cross ( albeit not its official headquar
ters , which are in rome ) . its main seminary , more ##au seminary , is loca
ted on the campus across st . joseph lake from the main building . old colleg
e , the oldest building on campus and located near the shore of st . mary lak
e , houses undergraduate seminar ##ians . retired priests and brothers reside
in fatima house ( a former retreat center ) , holy cross house , as well as c
ol ##umb ##a hall near the gr ##otto . the university through the more ##au s
eminary has ties to theologian frederick bu ##ech ##ner . while not catholic
 , bu ##ech ##ner has praised writers from notre dame and more ##au seminary
 created a bu ##ech ##ner prize for preaching . [SEP]
I0825 16:09:32.880713 140319166060416 run_squad.py:438] token_to_orig_map: 1
2:0 13:1 14:2 15:3 16:4 17:5 18:6 19:7 20:8 21:9 22:10 23:11 24:12 25:12 26:1
```

```
3 27:14 28:15 29:16 30:16 31:17 32:18 33:19 34:20 35:20 36:20 37:21 38:22 39:
23 40:23 41:24 42:24 43:25 44:25 45:26 46:27 47:28 48:29 49:30 50:31 51:32 5
2:32 53:33 54:34 55:35 56:36 57:37 58:38 59:38 60:39 61:40 62:40 63:41 64:42
 65:43 66:44 67:45 68:46 69:47 70:48 71:49 72:50 73:51 74:52 75:52 76:53 77:5
4 78:54 79:55 80:56 81:57 82:57 83:57 84:58 85:59 86:60 87:61 88:62 89:63 90:
64 91:65 92:66 93:66 94:67 95:68 96:69 97:69 98:69 99:70 100:71 101:72 102:72
103:73 104:74 105:75 106:76 107:76 108:76 109:77 110:78 111:79 112:80 113:80
 114:80 115:81 116:82 117:83 118:84 119:85 120:85 121:86 122:87 123:88 124:89
125:90 126:91 127:92 128:92 129:92 130:92 131:93 132:94 133:95 134:95 135:96
 136:96 137:96 138:97 139:98 140:99 141:100 142:101 143:102 144:103 145:104 1
46:104 147:105 148:106 149:107 150:108 151:108 152:108 153:109 154:110 155:11
1 156:111
I0825 16:09:32.881650 140319166060416 run_squad.py:440] token_is_max_context:
12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:Tr
ue 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:True 3
1:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 40:Tru
e 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:
True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:True
 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:T
rue 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:True 7
9:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 88:Tru
e 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:
True 99:True 100:True 101:True 102:True 103:True 104:True 105:True 106:True 1
07:True 108:True 109:True 110:True 111:True 112:True 113:True 114:True 115:Tr
ue 116:True 117:True 118:True 119:True 120:True 121:True 122:True 123:True 12
4:True 125:True 126:True 127:True 128:True 129:True 130:True 131:True 132:Tru
e 133:True 134:True 135:True 136:True 137:True 138:True 139:True 140:True 14
1:True 142:True 143:True 144:True 145:True 146:True 147:True 148:True 149:Tru
e 150:True 151:True 152:True 153:True 154:True 155:True 156:True
I0825 16:09:32.882565 140319166060416 run_squad.py:442] input_ids: 101 2054 3
633 2444 2012 27596 2160 2012 10289 8214 1029 102 1996 2118 2003 1996 2350 28
35 1997 1996 7769 1997 4151 2892 1006 12167 2025 2049 2880 4075 1010 2029 202
4 1999 4199 1007 1012 2049 2364 8705 1010 2062 4887 8705 1010 2003 2284 2006
 1996 3721 2408 2358 1012 3312 2697 2013 1996 2364 2311 1012 2214 2267 1010 1
996 4587 2311 2006 3721 1998 2284 2379 1996 5370 1997 2358 1012 2984 2697 101
0 3506 8324 18014 7066 1012 3394 8656 1998 3428 13960 1999 27596 2160 1006 10
37 2280 7822 2415 1007 1010 4151 2892 2160 1010 2004 2092 2004 8902 25438 205
0 2534 2379 1996 24665 23052 1012 1996 2118 2083 1996 2062 4887 8705 2038 720
8 2000 17200 5406 20934 15937 3678 1012 2096 2025 3234 1010 20934 15937 3678
 2038 5868 4898 2013 10289 8214 1998 2062 4887 8705 2580 1037 20934 15937 367
8 3396 2005 17979 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.883284 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.884641 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.885462 140319166060416 run_squad.py:451] start_position: 84
I0825 16:09:32.886071 140319166060416 run_squad.py:452] end_position: 87
I0825 16:09:32.886889 140319166060416 run_squad.py:454] answer: retired pries
ts and brothers
I0825 16:09:32.892927 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.894771 140319166060416 run_squad.py:432] unique_id: 1000000019
I0825 16:09:32.895796 140319166060416 run_squad.py:433] example_index: 14
I0825 16:09:32.896552 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.897982 140319166060416 run_squad.py:436] tokens: [CLS] which p
rize did frederick bu ##ech ##ner create ? [SEP] the university is the major
 seat of the congregation of holy cross ( albeit not its official headquarter
s , which are in rome ) . its main seminary , more ##au seminary , is located
on the campus across st . joseph lake from the main building . old college ,
 the oldest building on campus and located near the shore of st . mary lake ,
houses undergraduate seminar ##ians . retired priests and brothers reside in
 fatima house ( a former retreat center ) , holy cross house , as well as col
##umb ##a hall near the gr ##otto . the university through the more ##au semi
nary has ties to theologian frederick bu ##ech ##ner . while not catholic , b
u ##ech ##ner has praised writers from notre dame and more ##au seminary crea
ted a bu ##ech ##ner prize for preaching . [SEP]
I0825 16:09:32.899060 140319166060416 run_squad.py:438] token_to_orig_map: 1
1:0 12:1 13:2 14:3 15:4 16:5 17:6 18:7 19:8 20:9 21:10 22:11 23:12 24:12 25:1
3 26:14 27:15 28:16 29:16 30:17 31:18 32:19 33:20 34:20 35:20 36:21 37:22 38:
23 39:23 40:24 41:24 42:25 43:25 44:26 45:27 46:28 47:29 48:30 49:31 50:32 5
1:32 52:33 53:34 54:35 55:36 56:37 57:38 58:38 59:39 60:40 61:40 62:41 63:42
 64:43 65:44 66:45 67:46 68:47 69:48 70:49 71:50 72:51 73:52 74:52 75:53 76:5
4 77:54 78:55 79:56 80:57 81:57 82:57 83:58 84:59 85:60 86:61 87:62 88:63 89:
64 90:65 91:66 92:66 93:67 94:68 95:69 96:69 97:69 98:70 99:71 100:72 101:72
 102:73 103:74 104:75 105:76 106:76 107:76 108:77 109:78 110:79 111:80 112:80
113:80 114:81 115:82 116:83 117:84 118:85 119:85 120:86 121:87 122:88 123:89
 124:90 125:91 126:92 127:92 128:92 129:92 130:93 131:94 132:95 133:95 134:96
135:96 136:96 137:97 138:98 139:99 140:100 141:101 142:102 143:103 144:104 14
5:104 146:105 147:106 148:107 149:108 150:108 151:108 152:109 153:110 154:111
155:111
I0825 16:09:32.900388 140319166060416 run_squad.py:440] token_is_max_context:
11:True 12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:Tr
ue 21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:True 3
0:True 31:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 39:Tru
e 40:True 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:
True 50:True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:True
 59:True 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:T
rue 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:True 7
8:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 87:Tru
e 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:
True 98:True 99:True 100:True 101:True 102:True 103:True 104:True 105:True 10
6:True 107:True 108:True 109:True 110:True 111:True 112:True 113:True 114:Tru
e 115:True 116:True 117:True 118:True 119:True 120:True 121:True 122:True 12
3:True 124:True 125:True 126:True 127:True 128:True 129:True 130:True 131:Tru
e 132:True 133:True 134:True 135:True 136:True 137:True 138:True 139:True 14
0:True 141:True 142:True 143:True 144:True 145:True 146:True 147:True 148:Tru
e 149:True 150:True 151:True 152:True 153:True 154:True 155:True
I0825 16:09:32.902397 140319166060416 run_squad.py:442] input_ids: 101 2029 3
396 2106 5406 20934 15937 3678 3443 1029 102 1996 2118 2003 1996 2350 2835 19
97 1996 7769 1997 4151 2892 1006 12167 2025 2049 2880 4075 1010 2029 2024 199
9 4199 1007 1012 2049 2364 8705 1010 2062 4887 8705 1010 2003 2284 2006 1996
```

```
 3721 2408 2358 1012 3312 2697 2013 1996 2364 2311 1012 2214 2267 1010 1996 4
587 2311 2006 3721 1998 2284 2379 1996 5370 1997 2358 1012 2984 2697 1010 350
6 8324 18014 7066 1012 3394 8656 1998 3428 13960 1999 27596 2160 1006 1037 22
80 7822 2415 1007 1010 4151 2892 2160 1010 2004 2092 2004 8902 25438 2050 253
4 2379 1996 24665 23052 1012 1996 2118 2083 1996 2062 4887 8705 2038 7208 200
0 17200 5406 20934 15937 3678 1012 2096 2025 3234 1010 20934 15937 3678 2038
 5868 4898 2013 10289 8214 1998 2062 4887 8705 2580 1037 20934 15937 3678 339
6 2005 17979 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.903702 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.905030 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.906186 140319166060416 run_squad.py:451] start_position: 149
I0825 16:09:32.907162 140319166060416 run_squad.py:452] end_position: 154
I0825 16:09:32.907946 140319166060416 run_squad.py:454] answer: bu ##ech ##ne
r prize for preaching
I0825 16:09:32.913836 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.915213 140319166060416 run_squad.py:432] unique_id: 1000000020
I0825 16:09:32.916569 140319166060416 run_squad.py:433] example_index: 15
I0825 16:09:32.917253 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.918247 140319166060416 run_squad.py:436] tokens: [CLS] how man
y bs level degrees are offered in the college of engineering at notre dame ?
 [SEP] the college of engineering was established in 1920 , however , early c
ourses in civil and mechanical engineering were a part of the college of scie
nce since the 1870s . today the college , housed in the fitzpatrick , cu ##sh
ing , and st ##ins ##on - re ##mic ##k halls of engineering , includes five d
epartments of study – aerospace and mechanical engineering , chemical and bio
##mo ##le ##cular engineering , civil engineering and geological sciences , c
omputer science and engineering , and electrical engineering – with eight b .
s . degrees offered . additionally , the college offers five - year dual degr
ee programs with the colleges of arts and letters and of business awarding ad
ditional b . a . and master of business administration ( mba ) degrees , resp
ectively . [SEP]
I0825 16:09:32.919231 140319166060416 run_squad.py:438] token_to_orig_map: 1
8:0 19:1 20:2 21:3 22:4 23:5 24:6 25:7 26:7 27:8 28:8 29:9 30:10 31:11 32:12
 33:13 34:14 35:15 36:16 37:17 38:18 39:19 40:20 41:21 42:22 43:23 44:24 45:2
5 46:26 47:26 48:27 49:28 50:29 51:29 52:30 53:31 54:32 55:33 56:33 57:34 58:
34 59:34 60:35 61:36 62:36 63:36 64:36 65:36 66:36 67:36 68:37 69:38 70:39 7
1:39 72:40 73:41 74:42 75:43 76:44 77:45 78:46 79:47 80:48 81:49 82:49 83:50
 84:51 85:52 86:52 87:52 88:52 89:53 90:53 91:54 92:55 93:56 94:57 95:58 96:5
8 97:59 98:60 99:61 100:62 101:62 102:63 103:64 104:65 105:66 106:67 107:68 1
08:69 109:69 110:69 111:69 112:70 113:71 114:71 115:72 116:72 117:73 118:74 1
19:75 120:76 121:76 122:76 123:77 124:78 125:79 126:80 127:81 128:82 129:83 1
```

30:84 131:85 132:86 133:87 134:88 135:89 136:90 137:91 138:92 139:92 140:92 141:92 142:93 143:94 144:95 145:96 146:97 147:98 148:98 149:98 150:99 151:99 152:100 153:100
I0825 16:09:32.919951 140319166060416 run_squad.py:440] token_is_max_context: 18:True 19:True 20:True 21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:True 31:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 40:True 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:True 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:True 99:True 100:True 101:True 102:True 103:True 104:True 105:True 106:True 107:True 108:True 109:True 110:True 111:True 112:True 113:True 114:True 115:True 116:True 117:True 118:True 119:True 120:True 121:True 122:True 123:True 124:True 125:True 126:True 127:True 128:True 129:True 130:True 131:True 132:True 133:True 134:True 135:True 136:True 137:True 138:True 139:True 140:True 141:True 142:True 143:True 144:True 145:True 146:True 147:True 148:True 149:True 150:True 151:True 152:True 153:True
I0825 16:09:32.920889 140319166060416 run_squad.py:442] input_ids: 101 2129 2116 18667 2504 5445 2024 3253 1999 1996 2267 1997 3330 2012 10289 8214 1029 102 1996 2267 1997 3330 2001 2511 1999 4444 1010 2174 1010 2220 5352 1999 2942 1998 6228 3330 2020 1037 2112 1997 1996 2267 1997 2671 2144 1996 14896 1012 2651 1996 2267 1010 7431 1999 1996 26249 1010 12731 12227 1010 1998 2358 7076 2239 1011 2128 7712 2243 9873 1997 3330 1010 2950 2274 7640 1997 2817 1516 13395 1998 6228 3330 1010 5072 1998 16012 5302 2571 15431 3330 1010 2942 3330 1998 9843 4163 1010 3274 2671 1998 3330 1010 1998 5992 3330 1516 2007 2809 1038 1012 1055 1012 5445 3253 1012 5678 1010 1996 2267 4107 2274 1011 2095 7037 3014 3454 2007 1996 6667 1997 2840 1998 4144 1998 1997 2449 21467 3176 1038 1012 1037 1012 1998 3040 1997 2449 3447 1006 15038 1007 5445 1010 4414 1012 102 0
I0825 16:09:32.921759 140319166060416 run_squad.py:444] input_mask: 1 0
I0825 16:09:32.922456 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
I0825 16:09:32.923310 140319166060416 run_squad.py:451] start_position: 107
I0825 16:09:32.924293 140319166060416 run_squad.py:452] end_position: 107
I0825 16:09:32.925177 140319166060416 run_squad.py:454] answer: eight
I0825 16:09:32.931015 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.932420 140319166060416 run_squad.py:432] unique_id: 1000000021
I0825 16:09:32.933436 140319166060416 run_squad.py:433] example_index: 16
I0825 16:09:32.934528 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.935527 140319166060416 run_squad.py:436] tokens: [CLS] in what

```
year was the college of engineering at notre dame formed ? [SEP] the college
 of engineering was established in 1920 , however , early courses in civil an
d mechanical engineering were a part of the college of science since the 1870
s . today the college , housed in the fitzpatrick , cu ##shing , and st ##ins
##on - re ##mic ##k halls of engineering , includes five departments of study
– aerospace and mechanical engineering , chemical and bio ##mo ##le ##cular e
ngineering , civil engineering and geological sciences , computer science and
engineering , and electrical engineering – with eight b . s . degrees offered
. additionally , the college offers five - year dual degree programs with the
colleges of arts and letters and of business awarding additional b . a . and
 master of business administration ( mba ) degrees , respectively . [SEP]
I0825 16:09:32.936896 140319166060416 run_squad.py:438] token_to_orig_map: 1
5:0 16:1 17:2 18:3 19:4 20:5 21:6 22:7 23:7 24:8 25:8 26:9 27:10 28:11 29:12
 30:13 31:14 32:15 33:16 34:17 35:18 36:19 37:20 38:21 39:22 40:23 41:24 42:2
5 43:26 44:26 45:27 46:28 47:29 48:29 49:30 50:31 51:32 52:33 53:33 54:34 55:
34 56:34 57:35 58:36 59:36 60:36 61:36 62:36 63:36 64:36 65:37 66:38 67:39 6
8:39 69:40 70:41 71:42 72:43 73:44 74:45 75:46 76:47 77:48 78:49 79:49 80:50
 81:51 82:52 83:52 84:52 85:52 86:53 87:53 88:54 89:55 90:56 91:57 92:58 93:5
8 94:59 95:60 96:61 97:62 98:62 99:63 100:64 101:65 102:66 103:67 104:68 105:
69 106:69 107:69 108:69 109:70 110:71 111:71 112:72 113:72 114:73 115:74 116:
75 117:76 118:76 119:76 120:77 121:78 122:79 123:80 124:81 125:82 126:83 127:
84 128:85 129:86 130:87 131:88 132:89 133:90 134:91 135:92 136:92 137:92 138:
92 139:93 140:94 141:95 142:96 143:97 144:98 145:98 146:98 147:99 148:99 149:
100 150:100
I0825 16:09:32.937604 140319166060416 run_squad.py:440] token_is_max_context:
15:True 16:True 17:True 18:True 19:True 20:True 21:True 22:True 23:True 24:Tr
ue 25:True 26:True 27:True 28:True 29:True 30:True 31:True 32:True 33:True 3
4:True 35:True 36:True 37:True 38:True 39:True 40:True 41:True 42:True 43:Tru
e 44:True 45:True 46:True 47:True 48:True 49:True 50:True 51:True 52:True 53:
True 54:True 55:True 56:True 57:True 58:True 59:True 60:True 61:True 62:True
 63:True 64:True 65:True 66:True 67:True 68:True 69:True 70:True 71:True 72:T
rue 73:True 74:True 75:True 76:True 77:True 78:True 79:True 80:True 81:True 8
2:True 83:True 84:True 85:True 86:True 87:True 88:True 89:True 90:True 91:Tru
e 92:True 93:True 94:True 95:True 96:True 97:True 98:True 99:True 100:True 10
1:True 102:True 103:True 104:True 105:True 106:True 107:True 108:True 109:Tru
e 110:True 111:True 112:True 113:True 114:True 115:True 116:True 117:True 11
8:True 119:True 120:True 121:True 122:True 123:True 124:True 125:True 126:Tru
e 127:True 128:True 129:True 130:True 131:True 132:True 133:True 134:True 13
5:True 136:True 137:True 138:True 139:True 140:True 141:True 142:True 143:Tru
e 144:True 145:True 146:True 147:True 148:True 149:True 150:True
I0825 16:09:32.938586 140319166060416 run_squad.py:442] input_ids: 101 1999 2
054 2095 2001 1996 2267 1997 3330 2012 10289 8214 2719 1029 102 1996 2267 199
7 3330 2001 2511 1999 4444 1010 2174 1010 2220 5352 1999 2942 1998 6228 3330
 2020 1037 2112 1997 1996 2267 1997 2671 2144 1996 14896 1012 2651 1996 2267
 1010 7431 1999 1996 26249 1010 12731 12227 1010 1998 2358 7076 2239 1011 212
8 7712 2243 9873 1997 3330 1010 2950 2274 7640 1997 2817 1516 13395 1998 6228
3330 1010 5072 1998 16012 5302 2571 15431 3330 1010 2942 3330 1998 9843 4163
 1010 3274 2671 1998 3330 1010 1998 5992 3330 1516 2007 2809 1038 1012 1055 1
012 5445 3253 1012 5678 1010 1996 2267 4107 2274 1011 2095 7037 3014 3454 200
7 1996 6667 1997 2840 1998 4144 1998 1997 2449 21467 3176 1038 1012 1037 1012
1998 3040 1997 2449 3447 1006 15038 1007 5445 1010 4414 1012 102 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.939498 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.940192 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.941040 140319166060416 run_squad.py:451] start_position: 22
I0825 16:09:32.941675 140319166060416 run_squad.py:452] end_position: 22
I0825 16:09:32.942524 140319166060416 run_squad.py:454] answer: 1920
I0825 16:09:32.948590 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.949864 140319166060416 run_squad.py:432] unique_id: 1000000022
I0825 16:09:32.950973 140319166060416 run_squad.py:433] example_index: 17
I0825 16:09:32.952005 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.952842 140319166060416 run_squad.py:436] tokens: [CLS] before
 the creation of the college of engineering similar studies were carried out
 at which notre dame college ? [SEP] the college of engineering was establish
ed in 1920 , however , early courses in civil and mechanical engineering were
a part of the college of science since the 1870s . today the college , housed
in the fitzpatrick , cu ##shing , and st ##ins ##on - re ##mic ##k halls of e
ngineering , includes five departments of study – aerospace and mechanical en
gineering , chemical and bio ##mo ##le ##cular engineering , civil engineerin
g and geological sciences , computer science and engineering , and electrical
engineering – with eight b . s . degrees offered . additionally , the college
offers five - year dual degree programs with the colleges of arts and letters
and of business awarding additional b . a . and master of business administra
tion ( mba ) degrees , respectively . [SEP]
I0825 16:09:32.953942 140319166060416 run_squad.py:438] token_to_orig_map: 2
1:0 22:1 23:2 24:3 25:4 26:5 27:6 28:7 29:7 30:8 31:8 32:9 33:10 34:11 35:12
 36:13 37:14 38:15 39:16 40:17 41:18 42:19 43:20 44:21 45:22 46:23 47:24 48:2
5 49:26 50:26 51:27 52:28 53:29 54:29 55:30 56:31 57:32 58:33 59:33 60:34 61:
34 62:34 63:35 64:36 65:36 66:36 67:36 68:36 69:36 70:36 71:37 72:38 73:39 7
4:39 75:40 76:41 77:42 78:43 79:44 80:45 81:46 82:47 83:48 84:49 85:49 86:50
 87:51 88:52 89:52 90:52 91:52 92:53 93:53 94:54 95:55 96:56 97:57 98:58 99:5
8 100:59 101:60 102:61 103:62 104:62 105:63 106:64 107:65 108:66 109:67 110:6
8 111:69 112:69 113:69 114:69 115:70 116:71 117:71 118:72 119:72 120:73 121:7
4 122:75 123:76 124:76 125:76 126:77 127:78 128:79 129:80 130:81 131:82 132:8
3 133:84 134:85 135:86 136:87 137:88 138:89 139:90 140:91 141:92 142:92 143:9
2 144:92 145:93 146:94 147:95 148:96 149:97 150:98 151:98 152:98 153:99 154:9
9 155:100 156:100
I0825 16:09:32.954953 140319166060416 run_squad.py:440] token_is_max_context:
21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:Tr
ue 31:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 4
0:True 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:Tru
e 50:True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:
True 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True
 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:T
rue 79:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 8
8:True 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:Tru
e 98:True 99:True 100:True 101:True 102:True 103:True 104:True 105:True 106:T
rue 107:True 108:True 109:True 110:True 111:True 112:True 113:True 114:True 1
```

```
15:True 116:True 117:True 118:True 119:True 120:True 121:True 122:True 123:Tr
ue 124:True 125:True 126:True 127:True 128:True 129:True 130:True 131:True 13
2:True 133:True 134:True 135:True 136:True 137:True 138:True 139:True 140:Tru
e 141:True 142:True 143:True 144:True 145:True 146:True 147:True 148:True 14
9:True 150:True 151:True 152:True 153:True 154:True 155:True 156:True
I0825 16:09:32.956315 140319166060416 run_squad.py:442] input_ids: 101 2077 1
996 4325 1997 1996 2267 1997 3330 2714 2913 2020 3344 2041 2012 2029 10289 82
14 2267 1029 102 1996 2267 1997 3330 2001 2511 1999 4444 1010 2174 1010 2220
 5352 1999 2942 1998 6228 3330 2020 1037 2112 1997 1996 2267 1997 2671 2144 1
996 14896 1012 2651 1996 2267 1010 7431 1999 1996 26249 1010 12731 12227 1010
1998 2358 7076 2239 1011 2128 7712 2243 9873 1997 3330 1010 2950 2274 7640 19
97 2817 1516 13395 1998 6228 3330 1010 5072 1998 16012 5302 2571 15431 3330 1
010 2942 3330 1998 9843 4163 1010 3274 2671 1998 3330 1010 1998 5992 3330 151
6 2007 2809 1038 1012 1055 1012 5445 3253 1012 5678 1010 1996 2267 4107 2274
 1011 2095 7037 3014 3454 2007 1996 6667 1997 2840 1998 4144 1998 1997 2449 2
1467 3176 1038 1012 1037 1012 1998 3040 1997 2449 3447 1006 15038 1007 5445 1
010 4414 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.957097 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.958053 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.958717 140319166060416 run_squad.py:451] start_position: 43
I0825 16:09:32.959611 140319166060416 run_squad.py:452] end_position: 46
I0825 16:09:32.960250 140319166060416 run_squad.py:454] answer: the college o
f science
I0825 16:09:32.966176 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.968299 140319166060416 run_squad.py:432] unique_id: 1000000023
I0825 16:09:32.969269 140319166060416 run_squad.py:433] example_index: 18
I0825 16:09:32.970025 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.971028 140319166060416 run_squad.py:436] tokens: [CLS] how man
y departments are within the st ##ins ##on - re ##mic ##k hall of engineering
? [SEP] the college of engineering was established in 1920 , however , early
 courses in civil and mechanical engineering were a part of the college of sc
ience since the 1870s . today the college , housed in the fitzpatrick , cu ##
shing , and st ##ins ##on - re ##mic ##k halls of engineering , includes five
departments of study – aerospace and mechanical engineering , chemical and bi
o ##mo ##le ##cular engineering , civil engineering and geological sciences ,
computer science and engineering , and electrical engineering – with eight b
 . s . degrees offered . additionally , the college offers five - year dual d
egree programs with the colleges of arts and letters and of business awarding
additional b . a . and master of business administration ( mba ) degrees , re
spectively . [SEP]
I0825 16:09:32.971788 140319166060416 run_squad.py:438] token_to_orig_map: 1
```

```
9:0 20:1 21:2 22:3 23:4 24:5 25:6 26:7 27:7 28:8 29:8 30:9 31:10 32:11 33:12
 34:13 35:14 36:15 37:16 38:17 39:18 40:19 41:20 42:21 43:22 44:23 45:24 46:2
5 47:26 48:26 49:27 50:28 51:29 52:29 53:30 54:31 55:32 56:33 57:33 58:34 59:
34 60:34 61:35 62:36 63:36 64:36 65:36 66:36 67:36 68:36 69:37 70:38 71:39 7
2:39 73:40 74:41 75:42 76:43 77:44 78:45 79:46 80:47 81:48 82:49 83:49 84:50
 85:51 86:52 87:52 88:52 89:52 90:53 91:53 92:54 93:55 94:56 95:57 96:58 97:5
8 98:59 99:60 100:61 101:62 102:62 103:63 104:64 105:65 106:66 107:67 108:68
 109:69 110:69 111:69 112:69 113:70 114:71 115:71 116:72 117:72 118:73 119:74
120:75 121:76 122:76 123:76 124:77 125:78 126:79 127:80 128:81 129:82 130:83
 131:84 132:85 133:86 134:87 135:88 136:89 137:90 138:91 139:92 140:92 141:92
142:92 143:93 144:94 145:95 146:96 147:97 148:98 149:98 150:98 151:99 152:99
 153:100 154:100
I0825 16:09:32.972744 140319166060416 run_squad.py:440] token_is_max_context:
19:True 20:True 21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:Tr
ue 29:True 30:True 31:True 32:True 33:True 34:True 35:True 36:True 37:True 3
8:True 39:True 40:True 41:True 42:True 43:True 44:True 45:True 46:True 47:Tru
e 48:True 49:True 50:True 51:True 52:True 53:True 54:True 55:True 56:True 57:
True 58:True 59:True 60:True 61:True 62:True 63:True 64:True 65:True 66:True
 67:True 68:True 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:T
rue 77:True 78:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True 8
6:True 87:True 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:Tru
e 96:True 97:True 98:True 99:True 100:True 101:True 102:True 103:True 104:Tru
e 105:True 106:True 107:True 108:True 109:True 110:True 111:True 112:True 11
3:True 114:True 115:True 116:True 117:True 118:True 119:True 120:True 121:Tru
e 122:True 123:True 124:True 125:True 126:True 127:True 128:True 129:True 13
0:True 131:True 132:True 133:True 134:True 135:True 136:True 137:True 138:Tru
e 139:True 140:True 141:True 142:True 143:True 144:True 145:True 146:True 14
7:True 148:True 149:True 150:True 151:True 152:True 153:True 154:True
I0825 16:09:32.973737 140319166060416 run_squad.py:442] input_ids: 101 2129 2
116 7640 2024 2306 1996 2358 7076 2239 1011 2128 7712 2243 2534 1997 3330 102
9 102 1996 2267 1997 3330 2001 2511 1999 4444 1010 2174 1010 2220 5352 1999 2
942 1998 6228 3330 2020 1037 2112 1997 1996 2267 1997 2671 2144 1996 14896 10
12 2651 1996 2267 1010 7431 1999 1996 26249 1010 12731 12227 1010 1998 2358 7
076 2239 1011 2128 7712 2243 9873 1997 3330 1010 2950 2274 7640 1997 2817 151
6 13395 1998 6228 3330 1010 5072 1998 16012 5302 2571 15431 3330 1010 2942 33
30 1998 9843 4163 1010 3274 2671 1998 3330 1010 1998 5992 3330 1516 2007 2809
1038 1012 1055 1012 5445 3253 1012 5678 1010 1996 2267 4107 2274 1011 2095 70
37 3014 3454 2007 1996 6667 1997 2840 1998 4144 1998 1997 2449 21467 3176 103
8 1012 1037 1012 1998 3040 1997 2449 3447 1006 15038 1007 5445 1010 4414 1012
102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.974621 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.975363 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.976259 140319166060416 run_squad.py:451] start_position: 74
I0825 16:09:32.976906 140319166060416 run_squad.py:452] end_position: 74
I0825 16:09:32.977729 140319166060416 run_squad.py:454] answer: five
I0825 16:09:32.983318 140319166060416 run_squad.py:431] *** Example ***
I0825 16:09:32.984481 140319166060416 run_squad.py:432] unique_id: 1000000024
I0825 16:09:32.985460 140319166060416 run_squad.py:433] example_index: 19
I0825 16:09:32.986149 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:09:32.987067 140319166060416 run_squad.py:436] tokens: [CLS] the col
lege of science began to offer civil engineering courses beginning at what ti
me at notre dame ? [SEP] the college of engineering was established in 1920 ,
however , early courses in civil and mechanical engineering were a part of th
e college of science since the 1870s . today the college , housed in the fitz
patrick , cu ##shing , and st ##ins ##on - re ##mic ##k halls of engineering
 , includes five departments of study – aerospace and mechanical engineering
 , chemical and bio ##mo ##le ##cular engineering , civil engineering and geo
logical sciences , computer science and engineering , and electrical engineer
ing – with eight b . s . degrees offered . additionally , the college offers
 five - year dual degree programs with the colleges of arts and letters and o
f business awarding additional b . a . and master of business administration
 ( mba ) degrees , respectively . [SEP]
I0825 16:09:32.987752 140319166060416 run_squad.py:438] token_to_orig_map: 2
0:0 21:1 22:2 23:3 24:4 25:5 26:6 27:7 28:7 29:8 30:8 31:9 32:10 33:11 34:12
 35:13 36:14 37:15 38:16 39:17 40:18 41:19 42:20 43:21 44:22 45:23 46:24 47:2
5 48:26 49:26 50:27 51:28 52:29 53:29 54:30 55:31 56:32 57:33 58:33 59:34 60:
34 61:34 62:35 63:36 64:36 65:36 66:36 67:36 68:36 69:36 70:37 71:38 72:39 7
3:39 74:40 75:41 76:42 77:43 78:44 79:45 80:46 81:47 82:48 83:49 84:49 85:50
 86:51 87:52 88:52 89:52 90:52 91:53 92:53 93:54 94:55 95:56 96:57 97:58 98:5
8 99:59 100:60 101:61 102:62 103:62 104:63 105:64 106:65 107:66 108:67 109:68
110:69 111:69 112:69 113:69 114:70 115:71 116:71 117:72 118:72 119:73 120:74
 121:75 122:76 123:76 124:76 125:77 126:78 127:79 128:80 129:81 130:82 131:83
132:84 133:85 134:86 135:87 136:88 137:89 138:90 139:91 140:92 141:92 142:92
 143:92 144:93 145:94 146:95 147:96 148:97 149:98 150:98 151:98 152:99 153:99
154:100 155:100
I0825 16:09:32.988708 140319166060416 run_squad.py:440] token_is_max_context:
20:True 21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:Tr
ue 30:True 31:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 3
9:True 40:True 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:Tru
e 49:True 50:True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:
True 59:True 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True
 68:True 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:T
rue 78:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 8
7:True 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:Tru
e 97:True 98:True 99:True 100:True 101:True 102:True 103:True 104:True 105:Tr
ue 106:True 107:True 108:True 109:True 110:True 111:True 112:True 113:True 11
4:True 115:True 116:True 117:True 118:True 119:True 120:True 121:True 122:Tru
e 123:True 124:True 125:True 126:True 127:True 128:True 129:True 130:True 13
1:True 132:True 133:True 134:True 135:True 136:True 137:True 138:True 139:Tru
e 140:True 141:True 142:True 143:True 144:True 145:True 146:True 147:True 14
8:True 149:True 150:True 151:True 152:True 153:True 154:True 155:True
I0825 16:09:32.989616 140319166060416 run_squad.py:442] input_ids: 101 1996 2
267 1997 2671 2211 2000 3749 2942 3330 5352 2927 2012 2054 2051 2012 10289 82
14 1029 102 1996 2267 1997 3330 2001 2511 1999 4444 1010 2174 1010 2220 5352
 1999 2942 1998 6228 3330 2020 1037 2112 1997 1996 2267 1997 2671 2144 1996 1
4896 1012 2651 1996 2267 1010 7431 1999 1996 26249 1010 12731 12227 1010 1998
2358 7076 2239 1011 2128 7712 2243 9873 1997 3330 1010 2950 2274 7640 1997 28
17 1516 13395 1998 6228 3330 1010 5072 1998 16012 5302 2571 15431 3330 1010 2
```

```
942 3330 1998 9843 4163 1010 3274 2671 1998 3330 1010 1998 5992 3330 1516 200
7 2809 1038 1012 1055 1012 5445 3253 1012 5678 1010 1996 2267 4107 2274 1011
 2095 7037 3014 3454 2007 1996 6667 1997 2840 1998 4144 1998 1997 2449 21467
 3176 1038 1012 1037 1012 1998 3040 1997 2449 3447 1006 15038 1007 5445 1010
 4414 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.990648 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.991661 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:09:32.992510 140319166060416 run_squad.py:451] start_position: 47
I0825 16:09:32.993165 140319166060416 run_squad.py:452] end_position: 48
I0825 16:09:32.993778 140319166060416 run_squad.py:454] answer: the 1870s
```

In [26]:

```python
print(len(train_examples))
print(len(train_features))
```

```
87599
97077
```

In [27]:
```
print('***** Started training at {} *****'.format(datetime.datetime.now()))
print('  Num examples = {}'.format(len(train_examples)))
print('  Batch size = {}'.format(TRAIN_BATCH_SIZE))


tf.logging.info("  Num steps = %d", num_train_steps)
train_input_fn = run_squad.input_fn_builder(
    input_file=train_writer.filename,
    seq_length=MAX_SEQ_LENGTH,
    is_training=True,
    drop_remainder=True)

estimator.train(input_fn=train_input_fn, max_steps=num_train_steps)
print('***** Finished training at {} *****'.format(datetime.datetime.now()))
```

```
I0825 16:15:26.617880 140319166060416 <ipython-input-27-9544e7b43547>:6]   Nu
m steps = 10949
W0825 16:15:26.619843 140319166060416 deprecation_wrapper.py:119] From bert_r
epo/run_squad.py:691: The name tf.FixedLenFeature is deprecated. Please use t
f.io.FixedLenFeature instead.


***** Started training at 2019-08-25 16:15:26.616438 *****
  Num examples = 87599
  Batch size = 16
```

```
I0825 16:15:27.134990 140319166060416 tpu_system_metadata.py:78] Querying Ten
sorflow master (grpc://10.69.74.146:8470) for TPU system metadata.
I0825 16:15:27.154574 140319166060416 tpu_system_metadata.py:148] Found TPU s
ystem:
I0825 16:15:27.156421 140319166060416 tpu_system_metadata.py:149] *** Num TPU
Cores: 8
I0825 16:15:27.157664 140319166060416 tpu_system_metadata.py:150] *** Num TPU
Workers: 1
I0825 16:15:27.161403 140319166060416 tpu_system_metadata.py:152] *** Num TPU
Cores Per Worker: 8
I0825 16:15:27.163632 140319166060416 tpu_system_metadata.py:154] *** Availab
le Device: _DeviceAttributes(/job:worker/replica:0/task:0/device:CPU:0, CPU,
-1, 18165515893288348800)
I0825 16:15:27.172029 140319166060416 tpu_system_metadata.py:154] *** Availab
le Device: _DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:0, TPU,
17179869184, 5262853629327189391)
I0825 16:15:27.173218 140319166060416 tpu_system_metadata.py:154] *** Availab
le Device: _DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:1, TPU,
17179869184, 15522775392437584893)
I0825 16:15:27.176516 140319166060416 tpu_system_metadata.py:154] *** Availab
le Device: _DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:2, TPU,
17179869184, 863550791859807253)
I0825 16:15:27.178771 140319166060416 tpu_system_metadata.py:154] *** Availab
le Device: _DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:3, TPU,
17179869184, 15967457439584850702)
I0825 16:15:27.180298 140319166060416 tpu_system_metadata.py:154] *** Availab
le Device: _DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:4, TPU,
17179869184, 9825514860354380673)
I0825 16:15:27.181405 140319166060416 tpu_system_metadata.py:154] *** Availab
le Device: _DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:5, TPU,
17179869184, 9799457881871349772)
I0825 16:15:27.183599 140319166060416 tpu_system_metadata.py:154] *** Availab
le Device: _DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:6, TPU,
17179869184, 17961220308526691231)
I0825 16:15:27.184635 140319166060416 tpu_system_metadata.py:154] *** Availab
le Device: _DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:7, TPU,
17179869184, 4856695259171467612)
I0825 16:15:27.185744 140319166060416 tpu_system_metadata.py:154] *** Availab
le Device: _DeviceAttributes(/job:worker/replica:0/task:0/device:TPU_SYSTEM:
0, TPU_SYSTEM, 8589934592, 13923937947632350643)
I0825 16:15:27.187304 140319166060416 tpu_system_metadata.py:154] *** Availab
le Device: _DeviceAttributes(/job:worker/replica:0/task:0/device:XLA_CPU:0, X
LA_CPU, 17179869184, 8778042299146624752)
W0825 16:15:27.203457 140319166060416 deprecation.py:323] From /usr/local/li
b/python3.6/dist-packages/tensorflow/python/training/training_util.py:236: Va
riable.initialized_value (from tensorflow.python.ops.variables) is deprecated
and will be removed in a future version.
Instructions for updating:
Use Variable.read_value. Variables in 2.X are initialized automatically both
in eager and graph (inside tf.defun) contexts.
I0825 16:15:27.235837 140319166060416 estimator.py:1145] Calling model_fn.
W0825 16:15:27.276910 140319166060416 deprecation.py:323] From bert_repo/run_
squad.py:730: map_and_batch (from tensorflow.contrib.data.python.ops.batchin
g) is deprecated and will be removed in a future version.
Instructions for updating:
Use `tf.data.experimental.map_and_batch(...)`.
W0825 16:15:27.278246 140319166060416 deprecation.py:323] From /usr/local/li
```

```
b/python3.6/dist-packages/tensorflow/contrib/data/python/ops/batching.py:273:
map_and_batch (from tensorflow.python.data.experimental.ops.batching) is depr
ecated and will be removed in a future version.
Instructions for updating:
Use `tf.data.Dataset.map(map_func, num_parallel_calls)` followed by `tf.data.
Dataset.batch(batch_size, drop_remainder)`. Static tf.data optimizations will
take care of using the fused implementation.
W0825 16:15:27.282609 140319166060416 deprecation_wrapper.py:119] From bert_r
epo/run_squad.py:703: The name tf.parse_single_example is deprecated. Please
use tf.io.parse_single_example instead.

W0825 16:15:27.293697 140319166060416 deprecation.py:323] From bert_repo/run_
squad.py:710: to_int32 (from tensorflow.python.ops.math_ops) is deprecated an
d will be removed in a future version.
Instructions for updating:
Use `tf.cast` instead.
I0825 16:15:27.377592 140319166060416 run_squad.py:598] *** Features ***
I0825 16:15:27.379449 140319166060416 run_squad.py:600]   name = end_position
s, shape = (2,)
I0825 16:15:27.380994 140319166060416 run_squad.py:600]   name = input_ids, s
hape = (2, 256)
I0825 16:15:27.382128 140319166060416 run_squad.py:600]   name = input_mask,
shape = (2, 256)
I0825 16:15:27.383204 140319166060416 run_squad.py:600]   name = segment_ids,
shape = (2, 256)
I0825 16:15:27.384161 140319166060416 run_squad.py:600]   name = start_positi
ons, shape = (2,)
I0825 16:15:27.385141 140319166060416 run_squad.py:600]   name = unique_ids,
shape = (2,)
W0825 16:15:27.386635 140319166060416 deprecation_wrapper.py:119] From bert_r
epo/modeling.py:171: The name tf.variable_scope is deprecated. Please use tf.
compat.v1.variable_scope instead.

W0825 16:15:27.393050 140319166060416 deprecation_wrapper.py:119] From bert_r
epo/modeling.py:409: The name tf.get_variable is deprecated. Please use tf.co
mpat.v1.get_variable instead.

W0825 16:15:30.005829 140319166060416 deprecation_wrapper.py:119] From bert_r
epo/modeling.py:490: The name tf.assert_less_equal is deprecated. Please use
tf.compat.v1.assert_less_equal instead.

W0825 16:15:30.068442 140319166060416 deprecation.py:506] From bert_repo/mode
ling.py:358: calling dropout (from tensorflow.python.ops.nn_ops) with keep_pr
ob is deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - k
eep_prob`.
W0825 16:15:30.093674 140319166060416 deprecation.py:323] From bert_repo/mode
ling.py:671: dense (from tensorflow.python.layers.core) is deprecated and wil
l be removed in a future version.
Instructions for updating:
Use keras.layers.dense instead.
I0825 16:15:36.400019 140319166060416 run_squad.py:634] **** Trainable Variab
les ****
I0825 16:15:36.401609 140319166060416 run_squad.py:640]   name = bert/embeddi
ngs/word_embeddings:0, shape = (30522, 768), *INIT_FROM_CKPT*
I0825 16:15:36.402848 140319166060416 run_squad.py:640]   name = bert/embeddi
```

```
ngs/token_type_embeddings:0, shape = (2, 768), *INIT_FROM_CKPT*
I0825 16:15:36.403982 140319166060416 run_squad.py:640]   name = bert/embeddi
ngs/position_embeddings:0, shape = (512, 768), *INIT_FROM_CKPT*
I0825 16:15:36.404996 140319166060416 run_squad.py:640]   name = bert/embeddi
ngs/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.405969 140319166060416 run_squad.py:640]   name = bert/embeddi
ngs/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.406934 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.407829 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.408659 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.409580 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.410516 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.411376 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.412226 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:15:36.413080 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.413899 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.414664 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:15:36.415441 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:15:36.416197 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:15:36.416916 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:15:36.417729 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.418548 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.419309 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.420106 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.420848 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.421679 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.422506 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.423305 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.424203 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.425043 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:15:36.425869 140319166060416 run_squad.py:640]   name = bert/encode
```

```
r/layer_1/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.426631 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.427523 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:15:36.428313 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:15:36.429163 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:15:36.430076 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:15:36.430897 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.431692 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.432543 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.433368 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.434406 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.435254 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.436052 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.436774 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.437525 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.438300 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:15:36.439163 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.439937 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.440738 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:15:36.441615 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:15:36.442451 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:15:36.443281 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:15:36.444198 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.445058 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.446064 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.446927 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.447691 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.448477 140319166060416 run_squad.py:640]   name = bert/encode
```

```
r/layer_3/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.449312 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.450178 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.451012 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.451888 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:15:36.452671 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.453500 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.454319 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:15:36.455185 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:15:36.456156 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:15:36.457046 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:15:36.457882 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.458688 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.459537 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.460393 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.461219 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.462126 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.462968 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.463818 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.464696 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.465554 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:15:36.466371 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.467277 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.468105 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:15:36.468934 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:15:36.469830 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:15:36.470689 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
```

```
I0825 16:15:36.471585 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.472451 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.473287 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.474126 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.475020 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.475852 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.476722 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.477595 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.478415 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.479258 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:15:36.480123 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.480960 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.481819 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:15:36.482648 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:15:36.483526 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:15:36.484381 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:15:36.485224 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.486166 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.487032 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.487873 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.488758 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.489632 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.490537 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.491409 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.492250 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.493122 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:15:36.494014 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
```

```
I0825 16:15:36.494863 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.495748 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:15:36.496600 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:15:36.497454 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:15:36.498324 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:15:36.499187 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.500116 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.501082 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.501925 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.502794 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.503648 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.504512 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.505312 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.506186 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.507110 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:15:36.507986 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.508882 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.509755 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:15:36.510621 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:15:36.511478 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:15:36.512367 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:15:36.513208 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.514040 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.514876 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.515711 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.516599 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.517449 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
```

```
I0825 16:15:36.518322 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.519169 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.520162 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.521073 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:15:36.521922 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.522773 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.523658 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:15:36.524497 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:15:36.525375 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:15:36.526220 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:15:36.527123 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.528003 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.528854 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.529663 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.530590 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.531459 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.532361 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.533198 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.534049 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.534963 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:15:36.535845 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.536673 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.537588 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:15:36.538453 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:15:36.539285 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:15:36.540171 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:15:36.541025 140319166060416 run_squad.py:640]   name = bert/encode
```

```
r/layer_9/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.541928 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.542762 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.543592 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT
*
I0825 16:15:36.544445 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.545313 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.546193 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.547121 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT
*
I0825 16:15:36.547963 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.548759 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CK
PT*
I0825 16:15:36.549671 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.550549 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:15:36.551379 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKP
T*
I0825 16:15:36.552261 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:15:36.553109 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:15:36.553938 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:15:36.554833 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.555790 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.556643 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.557536 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT
*
I0825 16:15:36.558382 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.559193 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.560107 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.560952 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT
*
I0825 16:15:36.561838 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.562670 140319166060416 run_squad.py:640]   name = bert/encode
```

```
r/layer_11/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CK
PT*
I0825 16:15:36.563509 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.564343 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:15:36.565206 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKP
T*
I0825 16:15:36.566033 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:15:36.566926 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:15:36.567722 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:15:36.568694 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.569630 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.570486 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.571359 140319166060416 run_squad.py:640]   name = bert/pooler/
dense/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:15:36.572210 140319166060416 run_squad.py:640]   name = bert/pooler/
dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:15:36.573050 140319166060416 run_squad.py:640]   name = cls/squad/ou
tput_weights:0, shape = (2, 768)
I0825 16:15:36.573907 140319166060416 run_squad.py:640]   name = cls/squad/ou
tput_bias:0, shape = (2,)
W0825 16:15:36.618792 140319166060416 deprecation_wrapper.py:119] From bert_r
epo/optimization.py:27: The name tf.train.get_or_create_global_step is deprec
ated. Please use tf.compat.v1.train.get_or_create_global_step instead.

W0825 16:15:36.623750 140319166060416 deprecation_wrapper.py:119] From bert_r
epo/optimization.py:32: The name tf.train.polynomial_decay is deprecated. Ple
ase use tf.compat.v1.train.polynomial_decay instead.

W0825 16:15:36.642304 140319166060416 deprecation.py:323] From /usr/local/li
b/python3.6/dist-packages/tensorflow/python/keras/optimizer_v2/learning_rate_
schedule.py:409: div (from tensorflow.python.ops.math_ops) is deprecated and
will be removed in a future version.
Instructions for updating:
Deprecated in favor of operator or tf.math.divide.
W0825 16:15:37.238717 140319166060416 deprecation.py:323] From /usr/local/li
b/python3.6/dist-packages/tensorflow/python/ops/math_grad.py:1205: add_dispat
ch_support.<locals>.wrapper (from tensorflow.python.ops.array_ops) is depreca
ted and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
W0825 16:15:54.241975 140319166060416 deprecation_wrapper.py:119] From bert_r
epo/run_squad.py:627: The name tf.train.init_from_checkpoint is deprecated. P
lease use tf.compat.v1.train.init_from_checkpoint instead.

W0825 16:15:55.347643 140319166060416 deprecation_wrapper.py:119] From bert_r
epo/run_squad.py:628: The name tf.train.Scaffold is deprecated. Please use t
f.compat.v1.train.Scaffold instead.
```

```
I0825 16:15:56.369992 140319166060416 basic_session_run_hooks.py:541] Create
CheckpointSaverHook.
I0825 16:15:56.755395 140319166060416 estimator.py:1147] Done calling model_f
n.
I0825 16:15:59.731372 140319166060416 tpu_estimator.py:499] TPU job name work
er
I0825 16:16:03.716405 140319166060416 monitored_session.py:240] Graph was fin
alized.
W0825 16:16:03.765282 140319166060416 deprecation.py:323] From /usr/local/li
b/python3.6/dist-packages/tensorflow/python/training/saver.py:1276: checkpoin
t_exists (from tensorflow.python.training.checkpoint_management) is deprecate
d and will be removed in a future version.
Instructions for updating:
Use standard file APIs to check for files with this prefix.
I0825 16:16:03.810501 140319166060416 saver.py:1280] Restoring parameters fro
m gs://bert-on-squad/bert-checkpoints/models/SQUAD/model.ckpt-0
W0825 16:16:14.738395 140319166060416 deprecation.py:323] From /usr/local/li
b/python3.6/dist-packages/tensorflow/python/training/saver.py:1066: get_check
point_mtimes (from tensorflow.python.training.checkpoint_management) is depre
cated and will be removed in a future version.
Instructions for updating:
Use standard file utilities to get mtimes.
I0825 16:16:15.826352 140319166060416 session_manager.py:500] Running local_i
nit_op.
I0825 16:16:16.597252 140319166060416 session_manager.py:502] Done running lo
cal_init_op.
I0825 16:16:25.846445 140319166060416 basic_session_run_hooks.py:606] Saving
checkpoints for 0 into gs://bert-on-squad/bert-checkpoints/models/SQUAD/mode
l.ckpt.
W0825 16:16:48.678890 140319166060416 deprecation.py:323] From /usr/local/li
b/python3.6/dist-packages/tensorflow_estimator/python/estimator/tpu/tpu_estim
ator.py:741: Variable.load (from tensorflow.python.ops.variables) is deprecat
ed and will be removed in a future version.
Instructions for updating:
Prefer Variable.assign which has equivalent behavior in 2.X.
I0825 16:16:50.302542 140319166060416 util.py:98] Initialized dataset iterato
rs in 0 seconds
I0825 16:16:50.305584 140319166060416 session_support.py:332] Installing grac
eful shutdown hook.
I0825 16:16:50.313128 140319166060416 session_support.py:82] Creating heartbe
at manager for ['/job:worker/replica:0/task:0/device:CPU:0']
I0825 16:16:50.321530 140319166060416 session_support.py:105] Configuring wor
ker heartbeat: shutdown_mode: WAIT_FOR_COORDINATOR

I0825 16:16:50.330088 140319166060416 tpu_estimator.py:557] Init TPU system
I0825 16:16:57.765918 140319166060416 tpu_estimator.py:566] Initialized TPU i
n 7 seconds
I0825 16:16:57.768399 140316718171904 tpu_estimator.py:514] Starting infeed t
hread controller.
I0825 16:16:57.773868 140316690384640 tpu_estimator.py:533] Starting outfeed
thread controller.
I0825 16:16:58.553870 140319166060416 tpu_estimator.py:590] Enqueue next (100
0) batch(es) of data to infeed.
I0825 16:16:58.556298 140319166060416 tpu_estimator.py:594] Dequeue next (100
0) batch(es) of data from outfeed.
I0825 16:17:28.710566 140316690384640 tpu_estimator.py:275] Outfeed finished
```

```
for iteration (0, 0)
I0825 16:18:11.172186 140319166060416 basic_session_run_hooks.py:606] Saving
checkpoints for 1000 into gs://bert-on-squad/bert-checkpoints/models/SQUAD/mo
del.ckpt.
I0825 16:18:36.681888 140319166060416 basic_session_run_hooks.py:262] loss =
4.536075, step = 1000
I0825 16:18:36.685836 140319166060416 tpu_estimator.py:590] Enqueue next (100
0) batch(es) of data to infeed.
I0825 16:18:36.688987 140319166060416 tpu_estimator.py:594] Dequeue next (100
0) batch(es) of data from outfeed.
I0825 16:18:44.995945 140316690384640 tpu_estimator.py:275] Outfeed finished
for iteration (1, 0)
I0825 16:19:27.377380 140319166060416 basic_session_run_hooks.py:606] Saving
checkpoints for 2000 into gs://bert-on-squad/bert-checkpoints/models/SQUAD/mo
del.ckpt.
I0825 16:19:52.129538 140319166060416 basic_session_run_hooks.py:260] loss =
0.38595277, step = 2000 (75.448 sec)
I0825 16:19:52.132513 140319166060416 tpu_estimator.py:2159] global_step/sec:
13.2542
I0825 16:19:52.135496 140319166060416 tpu_estimator.py:2160] examples/sec: 21
2.067
I0825 16:19:52.138507 140319166060416 tpu_estimator.py:590] Enqueue next (100
0) batch(es) of data to infeed.
I0825 16:19:52.139749 140319166060416 tpu_estimator.py:594] Dequeue next (100
0) batch(es) of data from outfeed.
I0825 16:19:53.399676 140316690384640 tpu_estimator.py:275] Outfeed finished
for iteration (2, 0)
I0825 16:20:35.807610 140319166060416 basic_session_run_hooks.py:606] Saving
checkpoints for 3000 into gs://bert-on-squad/bert-checkpoints/models/SQUAD/mo
del.ckpt.
I0825 16:20:57.682199 140319166060416 basic_session_run_hooks.py:260] loss =
0.30558127, step = 3000 (65.553 sec)
I0825 16:20:57.690741 140319166060416 tpu_estimator.py:2159] global_step/sec:
15.2536
I0825 16:20:57.692292 140319166060416 tpu_estimator.py:2160] examples/sec: 24
4.058
I0825 16:20:57.699764 140319166060416 tpu_estimator.py:590] Enqueue next (100
0) batch(es) of data to infeed.
I0825 16:20:57.701190 140319166060416 tpu_estimator.py:594] Dequeue next (100
0) batch(es) of data from outfeed.
I0825 16:20:58.894317 140316690384640 tpu_estimator.py:275] Outfeed finished
for iteration (3, 0)
I0825 16:21:41.260964 140319166060416 basic_session_run_hooks.py:606] Saving
checkpoints for 4000 into gs://bert-on-squad/bert-checkpoints/models/SQUAD/mo
del.ckpt.
I0825 16:22:02.458211 140319166060416 basic_session_run_hooks.py:260] loss =
1.0719984, step = 4000 (64.776 sec)
I0825 16:22:02.460963 140319166060416 tpu_estimator.py:2159] global_step/sec:
15.4392
I0825 16:22:02.464643 140319166060416 tpu_estimator.py:2160] examples/sec: 24
7.027
I0825 16:22:02.467493 140319166060416 tpu_estimator.py:590] Enqueue next (100
0) batch(es) of data to infeed.
I0825 16:22:02.469333 140319166060416 tpu_estimator.py:594] Dequeue next (100
0) batch(es) of data from outfeed.
I0825 16:22:03.708269 140316690384640 tpu_estimator.py:275] Outfeed finished
for iteration (4, 0)
```

```
I0825 16:22:46.111944 140319166060416 basic_session_run_hooks.py:606] Saving
checkpoints for 5000 into gs://bert-on-squad/bert-checkpoints/models/SQUAD/mo
del.ckpt.
W0825 16:23:05.887398 140319166060416 deprecation.py:323] From /usr/local/li
b/python3.6/dist-packages/tensorflow/python/training/saver.py:960: remove_che
ckpoint (from tensorflow.python.training.checkpoint_management) is deprecated
and will be removed in a future version.
Instructions for updating:
Use standard file APIs to delete files with this prefix.
I0825 16:23:09.196741 140319166060416 basic_session_run_hooks.py:260] loss =
0.38893002, step = 5000 (66.739 sec)
I0825 16:23:09.204879 140319166060416 tpu_estimator.py:2159] global_step/sec:
14.9827
I0825 16:23:09.206665 140319166060416 tpu_estimator.py:2160] examples/sec: 23
9.723
I0825 16:23:09.211668 140319166060416 tpu_estimator.py:590] Enqueue next (100
0) batch(es) of data to infeed.
I0825 16:23:09.213521 140319166060416 tpu_estimator.py:594] Dequeue next (100
0) batch(es) of data from outfeed.
I0825 16:23:10.454283 140316690384640 tpu_estimator.py:275] Outfeed finished
for iteration (5, 0)
I0825 16:23:52.819438 140319166060416 basic_session_run_hooks.py:606] Saving
checkpoints for 6000 into gs://bert-on-squad/bert-checkpoints/models/SQUAD/mo
del.ckpt.
I0825 16:24:15.985026 140319166060416 basic_session_run_hooks.py:260] loss =
0.95278376, step = 6000 (66.788 sec)
I0825 16:24:15.989910 140319166060416 tpu_estimator.py:2159] global_step/sec:
14.9737
I0825 16:24:15.992052 140319166060416 tpu_estimator.py:2160] examples/sec: 23
9.579
I0825 16:24:15.995119 140319166060416 tpu_estimator.py:590] Enqueue next (100
0) batch(es) of data to infeed.
I0825 16:24:15.996619 140319166060416 tpu_estimator.py:594] Dequeue next (100
0) batch(es) of data from outfeed.
I0825 16:24:17.213131 140316690384640 tpu_estimator.py:275] Outfeed finished
for iteration (6, 0)
I0825 16:24:59.639587 140319166060416 basic_session_run_hooks.py:606] Saving
checkpoints for 7000 into gs://bert-on-squad/bert-checkpoints/models/SQUAD/mo
del.ckpt.
I0825 16:25:21.036397 140319166060416 basic_session_run_hooks.py:260] loss =
3.6535807, step = 7000 (65.051 sec)
I0825 16:25:21.044551 140319166060416 tpu_estimator.py:2159] global_step/sec:
15.3714
I0825 16:25:21.048130 140319166060416 tpu_estimator.py:2160] examples/sec: 24
5.942
I0825 16:25:21.052587 140319166060416 tpu_estimator.py:590] Enqueue next (100
0) batch(es) of data to infeed.
I0825 16:25:21.053680 140319166060416 tpu_estimator.py:594] Dequeue next (100
0) batch(es) of data from outfeed.
I0825 16:25:22.288265 140316690384640 tpu_estimator.py:275] Outfeed finished
for iteration (7, 0)
I0825 16:26:04.732192 140319166060416 basic_session_run_hooks.py:606] Saving
checkpoints for 8000 into gs://bert-on-squad/bert-checkpoints/models/SQUAD/mo
del.ckpt.
I0825 16:26:27.847544 140319166060416 basic_session_run_hooks.py:260] loss =
0.0018001954, step = 8000 (66.811 sec)
I0825 16:26:27.850726 140319166060416 tpu_estimator.py:2159] global_step/sec:
```

```
14.9687
I0825 16:26:27.853312 140319166060416 tpu_estimator.py:2160] examples/sec: 23
9.499
I0825 16:26:27.859663 140319166060416 tpu_estimator.py:590] Enqueue next (100
0) batch(es) of data to infeed.
I0825 16:26:27.860827 140319166060416 tpu_estimator.py:594] Dequeue next (100
0) batch(es) of data from outfeed.
I0825 16:26:29.110247 140316690384640 tpu_estimator.py:275] Outfeed finished
for iteration (8, 0)
I0825 16:27:11.475305 140319166060416 basic_session_run_hooks.py:606] Saving
checkpoints for 9000 into gs://bert-on-squad/bert-checkpoints/models/SQUAD/mo
del.ckpt.
I0825 16:27:36.668231 140319166060416 basic_session_run_hooks.py:260] loss =
2.2409706, step = 9000 (68.821 sec)
I0825 16:27:36.674966 140319166060416 tpu_estimator.py:2159] global_step/sec:
14.5298
I0825 16:27:36.676431 140319166060416 tpu_estimator.py:2160] examples/sec: 23
2.476
I0825 16:27:36.683845 140319166060416 tpu_estimator.py:590] Enqueue next (100
0) batch(es) of data to infeed.
I0825 16:27:36.685284 140319166060416 tpu_estimator.py:594] Dequeue next (100
0) batch(es) of data from outfeed.
I0825 16:27:37.911130 140316690384640 tpu_estimator.py:275] Outfeed finished
for iteration (9, 0)
I0825 16:28:20.292752 140319166060416 basic_session_run_hooks.py:606] Saving
checkpoints for 10000 into gs://bert-on-squad/bert-checkpoints/models/SQUAD/m
odel.ckpt.
I0825 16:28:46.344305 140319166060416 basic_session_run_hooks.py:260] loss =
0.7919205, step = 10000 (69.676 sec)
I0825 16:28:46.349406 140319166060416 tpu_estimator.py:2159] global_step/sec:
14.3525
I0825 16:28:46.351668 140319166060416 tpu_estimator.py:2160] examples/sec: 22
9.639
I0825 16:28:46.355234 140319166060416 tpu_estimator.py:590] Enqueue next (94
9) batch(es) of data to infeed.
I0825 16:28:46.356947 140319166060416 tpu_estimator.py:594] Dequeue next (94
9) batch(es) of data from outfeed.
I0825 16:28:47.613484 140316690384640 tpu_estimator.py:275] Outfeed finished
for iteration (10, 0)
I0825 16:29:27.877354 140319166060416 basic_session_run_hooks.py:260] loss =
0.0049015637, step = 10949 (41.533 sec)
I0825 16:29:27.881305 140319166060416 tpu_estimator.py:2159] global_step/sec:
22.8499
I0825 16:29:27.883299 140319166060416 tpu_estimator.py:2160] examples/sec: 36
5.599
I0825 16:29:27.886214 140319166060416 basic_session_run_hooks.py:606] Saving
checkpoints for 10949 into gs://bert-on-squad/bert-checkpoints/models/SQUAD/m
odel.ckpt.
I0825 16:29:52.389751 140319166060416 tpu_estimator.py:598] Stop infeed threa
d controller
I0825 16:29:52.391266 140319166060416 tpu_estimator.py:430] Shutting down Inf
eedController thread.
I0825 16:29:52.392169 140316718171904 tpu_estimator.py:425] InfeedController
received shutdown signal, stopping.
I0825 16:29:52.397272 140316718171904 tpu_estimator.py:530] Infeed thread fin
ished, shutting down.
I0825 16:29:52.404243 140319166060416 error_handling.py:96] infeed marked as
```

```
finished
I0825 16:29:52.405510 140319166060416 tpu_estimator.py:602] Stop output threa
d controller
I0825 16:29:52.407036 140319166060416 tpu_estimator.py:430] Shutting down Out
feedController thread.
I0825 16:29:52.409337 140316690384640 tpu_estimator.py:425] OutfeedController
received shutdown signal, stopping.
I0825 16:29:52.411226 140316690384640 tpu_estimator.py:541] Outfeed thread fi
nished, shutting down.
I0825 16:29:52.412441 140319166060416 error_handling.py:96] outfeed marked as
finished
I0825 16:29:52.413534 140319166060416 tpu_estimator.py:606] Shutdown TPU syst
em.
I0825 16:29:53.969046 140319166060416 estimator.py:368] Loss for final step:
0.0049015637.
I0825 16:29:53.971405 140319166060416 error_handling.py:96] training_loop mar
ked as finished

***** Finished training at 2019-08-25 16:29:53.972777 *****
```

Prediction

During prediction, we choose the best token span is taken that maximized start and probability.

In [0]:
```
eval_examples = read_squad_examples("/content/dev-v1.1.json",False)
```

In [0]:
```
eval_writer = run_squad.FeatureWriter(
    filename=os.path.join(OUTPUT_DIR, "eval.tf_record"),
    is_training=False)
```

In [0]:
```
def append_feature(feature):
    eval_features.append(feature)
    eval_writer.process_feature(feature)
```

In [31]:
```
eval_features = []

run_squad.convert_examples_to_features(
        examples=eval_examples,
        tokenizer=tokenizer,
        max_seq_length=MAX_SEQ_LENGTH,
        doc_stride=128,
        max_query_length=64,
        is_training=False,
        output_fn=append_feature)

eval_writer.close()
```

```
I0825 16:30:02.039860 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.041876 140319166060416 run_squad.py:432] unique_id: 1000000000
I0825 16:30:02.043239 140319166060416 run_squad.py:433] example_index: 0
I0825 16:30:02.044512 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.045924 140319166060416 run_squad.py:436] tokens: [CLS] which n
fl team represented the afc at super bowl 50 ? [SEP] super bowl 50 was an ame
rican football game to determine the champion of the national football league
( nfl ) for the 2015 season . the american football conference ( afc ) champi
on denver broncos defeated the national football conference ( nfc ) champion
carolina panthers 24 – 10 to earn their third super bowl title . the game was
played on february 7 , 2016 , at levi ' s stadium in the san francisco bay ar
ea at santa clara , california . as this was the 50th super bowl , the league
emphasized the " golden anniversary " with various gold - themed initiatives
, as well as temporarily suspend ##ing the tradition of naming each super bow
l game with roman nu ##meral ##s ( under which the game would have been known
as " super bowl l " ) , so that the logo could prominently feature the arabic
nu ##meral ##s 50 . [SEP]
I0825 16:30:02.047329 140319166060416 run_squad.py:438] token_to_orig_map: 1
3:0 14:1 15:2 16:3 17:4 18:5 19:6 20:7 21:8 22:9 23:10 24:11 25:12 26:13 27:1
4 28:15 29:16 30:17 31:17 32:17 33:18 34:19 35:20 36:21 37:21 38:22 39:23 40:
24 41:25 42:26 43:26 44:26 45:27 46:28 47:29 48:30 49:31 50:32 51:33 52:34 5
3:35 54:35 55:35 56:36 57:37 58:38 59:39 60:39 61:39 62:40 63:41 64:42 65:43
66:44 67:45 68:46 69:46 70:47 71:48 72:49 73:50 74:51 75:52 76:53 77:53 78:54
79:54 80:55 81:56 82:56 83:56 84:57 85:58 86:59 87:60 88:61 89:62 90:63 91:64
92:65 93:66 94:66 95:67 96:67 97:68 98:69 99:70 100:71 101:72 102:73 103:74 1
04:74 105:75 106:76 107:77 108:78 109:79 110:79 111:80 112:80 113:81 114:82 1
15:83 116:83 117:83 118:84 119:84 120:85 121:86 122:87 123:88 124:89 125:89 1
26:90 127:91 128:92 129:93 130:94 131:95 132:96 133:97 134:98 135:99 136:100
137:100 138:100 139:101 140:101 141:102 142:103 143:104 144:105 145:106 146:1
07 147:108 148:109 149:110 150:110 151:111 152:112 153:112 154:112 155:112 15
6:113 157:114 158:115 159:116 160:117 161:118 162:119 163:120 164:121 165:122
166:122 167:122 168:123 169:123
I0825 16:30:02.048746 140319166060416 run_squad.py:440] token_is_max_context:
13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:True 22:Tr
ue 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:True 31:True 3
2:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 40:True 41:Tru
e 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:True 51:
True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:True 60:True
61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:True 70:Tr
ue 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:True 79:True 8
0:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 88:True 89:Tru
e 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:True 99:
True 100:True 101:True 102:True 103:True 104:True 105:True 106:True 107:True
108:True 109:True 110:True 111:True 112:True 113:True 114:True 115:True 116:T
rue 117:True 118:True 119:True 120:True 121:True 122:True 123:True 124:True 1
25:True 126:True 127:True 128:True 129:True 130:True 131:True 132:True 133:Tr
ue 134:True 135:True 136:True 137:True 138:True 139:True 140:True 141:True 14
2:True 143:True 144:True 145:True 146:True 147:True 148:True 149:True 150:Tru
e 151:True 152:True 153:True 154:True 155:True 156:True 157:True 158:True 15
9:True 160:True 161:True 162:True 163:True 164:True 165:True 166:True 167:Tru
e 168:True 169:True
I0825 16:30:02.050205 140319166060416 run_squad.py:442] input_ids: 101 2029 5
088 2136 3421 1996 10511 2012 3565 4605 2753 1029 102 3565 4605 2753 2001 201
9 2137 2374 2208 2000 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007
2005 1996 2325 2161 1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169
3249 1996 2120 2374 3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000
7796 2037 2353 3565 4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 23
```

```
55 1010 2012 11902 1005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 102
54 1010 2662 1012 2004 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 19
96 1000 3585 5315 1000 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 81
84 28324 2075 1996 4535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 
2015 1006 2104 2029 1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 10
00 1007 1010 2061 2008 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2
753 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.051340 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.052436 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.060137 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.061613 140319166060416 run_squad.py:432] unique_id: 1000000001
I0825 16:30:02.062654 140319166060416 run_squad.py:433] example_index: 1
I0825 16:30:02.063781 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.064887 140319166060416 run_squad.py:436] tokens: [CLS] which n
fl team represented the nfc at super bowl 50 ? [SEP] super bowl 50 was an ame
rican football game to determine the champion of the national football league
( nfl ) for the 2015 season . the american football conference ( afc ) champi
on denver broncos defeated the national football conference ( nfc ) champion 
carolina panthers 24 – 10 to earn their third super bowl title . the game was
played on february 7 , 2016 , at levi ' s stadium in the san francisco bay ar
ea at santa clara , california . as this was the 50th super bowl , the league
emphasized the " golden anniversary " with various gold - themed initiatives 
, as well as temporarily suspend ##ing the tradition of naming each super bow
l game with roman nu ##meral ##s ( under which the game would have been known
as " super bowl l " ) , so that the logo could prominently feature the arabic
nu ##meral ##s 50 . [SEP]
I0825 16:30:02.066072 140319166060416 run_squad.py:438] token_to_orig_map: 1
3:0 14:1 15:2 16:3 17:4 18:5 19:6 20:7 21:8 22:9 23:10 24:11 25:12 26:13 27:1
4 28:15 29:16 30:17 31:17 32:17 33:18 34:19 35:20 36:21 37:21 38:22 39:23 40:
24 41:25 42:26 43:26 44:26 45:27 46:28 47:29 48:30 49:31 50:32 51:33 52:34 5
3:35 54:35 55:35 56:36 57:37 58:38 59:39 60:39 61:39 62:40 63:41 64:42 65:43
66:44 67:45 68:46 69:46 70:47 71:48 72:49 73:50 74:51 75:52 76:53 77:53 78:54
79:54 80:55 81:56 82:56 83:56 84:57 85:58 86:59 87:60 88:61 89:62 90:63 91:64
92:65 93:66 94:66 95:67 96:67 97:68 98:69 99:70 100:71 101:72 102:73 103:74 1
04:74 105:75 106:76 107:77 108:78 109:79 110:79 111:80 112:80 113:81 114:82 1
15:83 116:83 117:83 118:84 119:84 120:85 121:86 122:87 123:88 124:89 125:89 1
26:90 127:91 128:92 129:93 130:94 131:95 132:96 133:97 134:98 135:99 136:100 
137:100 138:100 139:101 140:101 141:102 142:103 143:104 144:105 145:106 146:1
07 147:108 148:109 149:110 150:110 151:111 152:112 153:112 154:112 155:112 15
6:113 157:114 158:115 159:116 160:117 161:118 162:119 163:120 164:121 165:122
166:122 167:122 168:123 169:123
```

```
I0825 16:30:02.067120 140319166060416 run_squad.py:440] token_is_max_context:
13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:True 22:Tr
ue 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:True 31:True 3
2:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 40:True 41:Tru
e 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:True 51:
True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:True 60:True
61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:True 70:Tr
ue 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:True 79:True 8
0:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 88:True 89:Tru
e 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:True 99:
True 100:True 101:True 102:True 103:True 104:True 105:True 106:True 107:True
108:True 109:True 110:True 111:True 112:True 113:True 114:True 115:True 116:T
rue 117:True 118:True 119:True 120:True 121:True 122:True 123:True 124:True 1
25:True 126:True 127:True 128:True 129:True 130:True 131:True 132:True 133:Tr
ue 134:True 135:True 136:True 137:True 138:True 139:True 140:True 141:True 14
2:True 143:True 144:True 145:True 146:True 147:True 148:True 149:True 150:Tru
e 151:True 152:True 153:True 154:True 155:True 156:True 157:True 158:True 15
9:True 160:True 161:True 162:True 163:True 164:True 165:True 166:True 167:Tru
e 168:True 169:True
I0825 16:30:02.068316 140319166060416 run_squad.py:442] input_ids: 101 2029 5
088 2136 3421 1996 22309 2012 3565 4605 2753 1029 102 3565 4605 2753 2001 201
9 2137 2374 2208 2000 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007
2005 1996 2325 2161 1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169
3249 1996 2120 2374 3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000
7796 2037 2353 3565 4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 23
55 1010 2012 11902 1005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 102
54 1010 2662 1012 2004 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 19
96 1000 3585 5315 1000 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 81
84 28324 2075 1996 4535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990
2015 1006 2104 2029 1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 10
00 1007 1010 2061 2008 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2
753 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.069406 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.070514 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.078215 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.079687 140319166060416 run_squad.py:432] unique_id: 1000000002
I0825 16:30:02.080738 140319166060416 run_squad.py:433] example_index: 2
I0825 16:30:02.081819 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.082911 140319166060416 run_squad.py:436] tokens: [CLS] where d
id super bowl 50 take place ? [SEP] super bowl 50 was an american football ga
me to determine the champion of the national football league ( nfl ) for the
```

```
2015 season . the american football conference ( afc ) champion denver bronco
s defeated the national football conference ( nfc ) champion carolina panther
s 24 – 10 to earn their third super bowl title . the game was played on febru
ary 7 , 2016 , at levi ' s stadium in the san francisco bay area at santa cla
ra , california . as this was the 50th super bowl , the league emphasized the
" golden anniversary " with various gold - themed initiatives , as well as te
mporarily suspend ##ing the tradition of naming each super bowl game with rom
an nu ##meral ##s ( under which the game would have been known as " super bow
l l " ) , so that the logo could prominently feature the arabic nu ##meral ##
s 50 . [SEP]
I0825 16:30:02.084084 140319166060416 run_squad.py:438] token_to_orig_map: 1
0:0 11:1 12:2 13:3 14:4 15:5 16:6 17:7 18:8 19:9 20:10 21:11 22:12 23:13 24:1
4 25:15 26:16 27:17 28:17 29:17 30:18 31:19 32:20 33:21 34:21 35:22 36:23 37:
24 38:25 39:26 40:26 41:26 42:27 43:28 44:29 45:30 46:31 47:32 48:33 49:34 5
0:35 51:35 52:35 53:36 54:37 55:38 56:39 57:39 58:39 59:40 60:41 61:42 62:43
63:44 64:45 65:46 66:46 67:47 68:48 69:49 70:50 71:51 72:52 73:53 74:53 75:54
76:54 77:55 78:56 79:56 80:56 81:57 82:58 83:59 84:60 85:61 86:62 87:63 88:64
89:65 90:66 91:66 92:67 93:67 94:68 95:69 96:70 97:71 98:72 99:73 100:74 101:
74 102:75 103:76 104:77 105:78 106:79 107:79 108:80 109:80 110:81 111:82 112:
83 113:83 114:83 115:84 116:84 117:85 118:86 119:87 120:88 121:89 122:89 123:
90 124:91 125:92 126:93 127:94 128:95 129:96 130:97 131:98 132:99 133:100 13
4:100 135:100 136:101 137:101 138:102 139:103 140:104 141:105 142:106 143:107
144:108 145:109 146:110 147:110 148:111 149:112 150:112 151:112 152:112 153:1
13 154:114 155:115 156:116 157:117 158:118 159:119 160:120 161:121 162:122 16
3:122 164:122 165:123 166:123
I0825 16:30:02.085136 140319166060416 run_squad.py:440] token_is_max_context:
10:True 11:True 12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:Tr
ue 20:True 21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:True 2
9:True 30:True 31:True 32:True 33:True 34:True 35:True 36:True 37:True 38:Tru
e 39:True 40:True 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:
True 49:True 50:True 51:True 52:True 53:True 54:True 55:True 56:True 57:True
58:True 59:True 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:Tr
ue 68:True 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:True 7
7:True 78:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True 86:Tru
e 87:True 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:
True 97:True 98:True 99:True 100:True 101:True 102:True 103:True 104:True 10
5:True 106:True 107:True 108:True 109:True 110:True 111:True 112:True 113:Tru
e 114:True 115:True 116:True 117:True 118:True 119:True 120:True 121:True 12
2:True 123:True 124:True 125:True 126:True 127:True 128:True 129:True 130:Tru
e 131:True 132:True 133:True 134:True 135:True 136:True 137:True 138:True 13
9:True 140:True 141:True 142:True 143:True 144:True 145:True 146:True 147:Tru
e 148:True 149:True 150:True 151:True 152:True 153:True 154:True 155:True 15
6:True 157:True 158:True 159:True 160:True 161:True 162:True 163:True 164:Tru
e 165:True 166:True
I0825 16:30:02.086303 140319166060416 run_squad.py:442] input_ids: 101 2073 2
106 3565 4605 2753 2202 2173 1029 102 3565 4605 2753 2001 2019 2137 2374 2208
2000 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 2005 1996 2325 21
61 1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169 3249 1996 2120 23
74 3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000 7796 2037 2353 35
65 4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355 1010 2012 1190
2 1005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254 1010 2662 1012
2004 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 1996 1000 3585 5315
1000 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184 28324 2075 1996
4535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 2015 1006 2104 2029
1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000 1007 1010 2061 20
08 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2753 1012 102 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0
I0825 16:30:02.087487 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.088594 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.096409 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.097949 140319166060416 run_squad.py:432] unique_id: 1000000003
I0825 16:30:02.099088 140319166060416 run_squad.py:433] example_index: 3
I0825 16:30:02.100185 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.101405 140319166060416 run_squad.py:436] tokens: [CLS] which n
fl team won super bowl 50 ? [SEP] super bowl 50 was an american football game
to determine the champion of the national football league ( nfl ) for the 201
5 season . the american football conference ( afc ) champion denver broncos d
efeated the national football conference ( nfc ) champion carolina panthers 2
4 – 10 to earn their third super bowl title . the game was played on february
7 , 2016 , at levi ' s stadium in the san francisco bay area at santa clara ,
california . as this was the 50th super bowl , the league emphasized the " go
lden anniversary " with various gold - themed initiatives , as well as tempor
arily suspend ##ing the tradition of naming each super bowl game with roman n
u ##meral ##s ( under which the game would have been known as " super bowl l
" ) , so that the logo could prominently feature the arabic nu ##meral ##s 50
. [SEP]
I0825 16:30:02.102592 140319166060416 run_squad.py:438] token_to_orig_map: 1
0:0 11:1 12:2 13:3 14:4 15:5 16:6 17:7 18:8 19:9 20:10 21:11 22:12 23:13 24:1
4 25:15 26:16 27:17 28:17 29:17 30:18 31:19 32:20 33:21 34:21 35:22 36:23 37:
24 38:25 39:26 40:26 41:26 42:27 43:28 44:29 45:30 46:31 47:32 48:33 49:34 5
0:35 51:35 52:35 53:36 54:37 55:38 56:39 57:39 58:39 59:40 60:41 61:42 62:43
63:44 64:45 65:46 66:46 67:47 68:48 69:49 70:50 71:51 72:52 73:53 74:53 75:54
76:54 77:55 78:56 79:56 80:56 81:57 82:58 83:59 84:60 85:61 86:62 87:63 88:64
89:65 90:66 91:66 92:67 93:67 94:68 95:69 96:70 97:71 98:72 99:73 100:74 101:
74 102:75 103:76 104:77 105:78 106:79 107:79 108:80 109:80 110:81 111:82 112:
83 113:83 114:83 115:84 116:84 117:85 118:86 119:87 120:88 121:89 122:89 123:
90 124:91 125:92 126:93 127:94 128:95 129:96 130:97 131:98 132:99 133:100 13
4:100 135:100 136:101 137:101 138:102 139:103 140:104 141:105 142:106 143:107
144:108 145:109 146:110 147:110 148:111 149:112 150:112 151:112 152:112 153:1
13 154:114 155:115 156:116 157:117 158:118 159:119 160:120 161:121 162:122 16
3:122 164:122 165:123 166:123
I0825 16:30:02.103817 140319166060416 run_squad.py:440] token_is_max_context:
10:True 11:True 12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:Tr
ue 20:True 21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:True 2
9:True 30:True 31:True 32:True 33:True 34:True 35:True 36:True 37:True 38:Tru
e 39:True 40:True 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:
True 49:True 50:True 51:True 52:True 53:True 54:True 55:True 56:True 57:True
58:True 59:True 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:Tr
```

```
ue 68:True 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:True 7
7:True 78:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True 86:Tru
e 87:True 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:
True 97:True 98:True 99:True 100:True 101:True 102:True 103:True 104:True 10
5:True 106:True 107:True 108:True 109:True 110:True 111:True 112:True 113:Tru
e 114:True 115:True 116:True 117:True 118:True 119:True 120:True 121:True 12
2:True 123:True 124:True 125:True 126:True 127:True 128:True 129:True 130:Tru
e 131:True 132:True 133:True 134:True 135:True 136:True 137:True 138:True 13
9:True 140:True 141:True 142:True 143:True 144:True 145:True 146:True 147:Tru
e 148:True 149:True 150:True 151:True 152:True 153:True 154:True 155:True 15
6:True 157:True 158:True 159:True 160:True 161:True 162:True 163:True 164:Tru
e 165:True 166:True
I0825 16:30:02.104994 140319166060416 run_squad.py:442] input_ids: 101 2029 5
088 2136 2180 3565 4605 2753 1029 102 3565 4605 2753 2001 2019 2137 2374 2208
2000 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 2005 1996 2325 21
61 1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169 3249 1996 2120 23
74 3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000 7796 2037 2353 35
65 4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355 1010 2012 1190
2 1005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254 1010 2662 1012
2004 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 1996 1000 3585 5315
1000 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184 28324 2075 1996
4535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 2015 1006 2104 2029
1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000 1007 1010 2061 20
08 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2753 1012 102 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0
I0825 16:30:02.106067 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.107072 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.115176 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.116714 140319166060416 run_squad.py:432] unique_id: 1000000004
I0825 16:30:02.117755 140319166060416 run_squad.py:433] example_index: 4
I0825 16:30:02.118696 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.119671 140319166060416 run_squad.py:436] tokens: [CLS] what co
lor was used to emphasize the 50th anniversary of the super bowl ? [SEP] supe
r bowl 50 was an american football game to determine the champion of the nati
onal football league ( nfl ) for the 2015 season . the american football conf
erence ( afc ) champion denver broncos defeated the national football confere
nce ( nfc ) champion carolina panthers 24 – 10 to earn their third super bowl
title . the game was played on february 7 , 2016 , at levi ' s stadium in the
san francisco bay area at santa clara , california . as this was the 50th sup
er bowl , the league emphasized the " golden anniversary " with various gold
- themed initiatives , as well as temporarily suspend ##ing the tradition of
```

```
naming each super bowl game with roman nu ##meral ##s ( under which the game
would have been known as " super bowl l " ) , so that the logo could prominen
tly feature the arabic nu ##meral ##s 50 . [SEP]
I0825 16:30:02.120650 140319166060416 run_squad.py:438] token_to_orig_map: 1
6:0 17:1 18:2 19:3 20:4 21:5 22:6 23:7 24:8 25:9 26:10 27:11 28:12 29:13 30:1
4 31:15 32:16 33:17 34:17 35:17 36:18 37:19 38:20 39:21 40:21 41:22 42:23 43:
24 44:25 45:26 46:26 47:26 48:27 49:28 50:29 51:30 52:31 53:32 54:33 55:34 5
6:35 57:35 58:35 59:36 60:37 61:38 62:39 63:39 64:39 65:40 66:41 67:42 68:43
69:44 70:45 71:46 72:46 73:47 74:48 75:49 76:50 77:51 78:52 79:53 80:53 81:54
82:54 83:55 84:56 85:56 86:56 87:57 88:58 89:59 90:60 91:61 92:62 93:63 94:64
95:65 96:66 97:66 98:67 99:67 100:68 101:69 102:70 103:71 104:72 105:73 106:7
4 107:74 108:75 109:76 110:77 111:78 112:79 113:79 114:80 115:80 116:81 117:8
2 118:83 119:83 120:83 121:84 122:84 123:85 124:86 125:87 126:88 127:89 128:8
9 129:90 130:91 131:92 132:93 133:94 134:95 135:96 136:97 137:98 138:99 139:1
00 140:100 141:100 142:101 143:101 144:102 145:103 146:104 147:105 148:106 14
9:107 150:108 151:109 152:110 153:110 154:111 155:112 156:112 157:112 158:112
159:113 160:114 161:115 162:116 163:117 164:118 165:119 166:120 167:121 168:1
22 169:122 170:122 171:123 172:123
I0825 16:30:02.121656 140319166060416 run_squad.py:440] token_is_max_context:
16:True 17:True 18:True 19:True 20:True 21:True 22:True 23:True 24:True 25:Tr
ue 26:True 27:True 28:True 29:True 30:True 31:True 32:True 33:True 34:True 3
5:True 36:True 37:True 38:True 39:True 40:True 41:True 42:True 43:True 44:Tru
e 45:True 46:True 47:True 48:True 49:True 50:True 51:True 52:True 53:True 54:
True 55:True 56:True 57:True 58:True 59:True 60:True 61:True 62:True 63:True
64:True 65:True 66:True 67:True 68:True 69:True 70:True 71:True 72:True 73:Tr
ue 74:True 75:True 76:True 77:True 78:True 79:True 80:True 81:True 82:True 8
3:True 84:True 85:True 86:True 87:True 88:True 89:True 90:True 91:True 92:Tru
e 93:True 94:True 95:True 96:True 97:True 98:True 99:True 100:True 101:True 1
02:True 103:True 104:True 105:True 106:True 107:True 108:True 109:True 110:Tr
ue 111:True 112:True 113:True 114:True 115:True 116:True 117:True 118:True 11
9:True 120:True 121:True 122:True 123:True 124:True 125:True 126:True 127:Tru
e 128:True 129:True 130:True 131:True 132:True 133:True 134:True 135:True 13
6:True 137:True 138:True 139:True 140:True 141:True 142:True 143:True 144:Tru
e 145:True 146:True 147:True 148:True 149:True 150:True 151:True 152:True 15
3:True 154:True 155:True 156:True 157:True 158:True 159:True 160:True 161:Tru
e 162:True 163:True 164:True 165:True 166:True 167:True 168:True 169:True 17
0:True 171:True 172:True
I0825 16:30:02.122686 140319166060416 run_squad.py:442] input_ids: 101 2054 3
609 2001 2109 2000 17902 1996 12951 5315 1997 1996 3565 4605 1029 102 3565 46
05 2753 2001 2019 2137 2374 2208 2000 5646 1996 3410 1997 1996 2120 2374 2223
1006 5088 1007 2005 1996 2325 2161 1012 1996 2137 2374 3034 1006 10511 1007 3
410 7573 14169 3249 1996 2120 2374 3034 1006 22309 1007 3410 3792 12915 2484
1516 2184 2000 7796 2037 2353 3565 4605 2516 1012 1996 2208 2001 2209 2006 23
37 1021 1010 2355 1010 2012 11902 1005 1055 3346 1999 1996 2624 3799 3016 218
1 2012 4203 10254 1010 2662 1012 2004 2023 2001 1996 12951 3565 4605 1010 199
6 2223 13155 1996 1000 3585 5315 1000 2007 2536 2751 1011 11773 11107 1010 20
04 2092 2004 8184 28324 2075 1996 4535 1997 10324 2169 3565 4605 2208 2007 31
42 16371 28990 2015 1006 2104 2029 1996 2208 2052 2031 2042 2124 2004 1000 35
65 4605 1048 1000 1007 1010 2061 2008 1996 8154 2071 14500 3444 1996 5640 163
71 28990 2015 2753 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.123662 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.124694 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.134126 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.135489 140319166060416 run_squad.py:432] unique_id: 1000000005
I0825 16:30:02.136541 140319166060416 run_squad.py:433] example_index: 5
I0825 16:30:02.137623 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.138623 140319166060416 run_squad.py:436] tokens: [CLS] what wa
s the theme of super bowl 50 ? [SEP] super bowl 50 was an american football g
ame to determine the champion of the national football league ( nfl ) for the
2015 season . the american football conference ( afc ) champion denver bronco
s defeated the national football conference ( nfc ) champion carolina panther
s 24 – 10 to earn their third super bowl title . the game was played on febru
ary 7 , 2016 , at levi ' s stadium in the san francisco bay area at santa cla
ra , california . as this was the 50th super bowl , the league emphasized the
" golden anniversary " with various gold - themed initiatives , as well as te
mporarily suspend ##ing the tradition of naming each super bowl game with rom
an nu ##meral ##s ( under which the game would have been known as " super bow
l l " ) , so that the logo could prominently feature the arabic nu ##meral ##
s 50 . [SEP]
I0825 16:30:02.139687 140319166060416 run_squad.py:438] token_to_orig_map: 1
1:0 12:1 13:2 14:3 15:4 16:5 17:6 18:7 19:8 20:9 21:10 22:11 23:12 24:13 25:1
4 26:15 27:16 28:17 29:17 30:17 31:18 32:19 33:20 34:21 35:21 36:22 37:23 38:
24 39:25 40:26 41:26 42:26 43:27 44:28 45:29 46:30 47:31 48:32 49:33 50:34 5
1:35 52:35 53:35 54:36 55:37 56:38 57:39 58:39 59:39 60:40 61:41 62:42 63:43
64:44 65:45 66:46 67:46 68:47 69:48 70:49 71:50 72:51 73:52 74:53 75:53 76:54
77:54 78:55 79:56 80:56 81:56 82:57 83:58 84:59 85:60 86:61 87:62 88:63 89:64
90:65 91:66 92:66 93:67 94:67 95:68 96:69 97:70 98:71 99:72 100:73 101:74 10
2:74 103:75 104:76 105:77 106:78 107:79 108:79 109:80 110:80 111:81 112:82 11
3:83 114:83 115:83 116:84 117:84 118:85 119:86 120:87 121:88 122:89 123:89 12
4:90 125:91 126:92 127:93 128:94 129:95 130:96 131:97 132:98 133:99 134:100 1
35:100 136:100 137:101 138:101 139:102 140:103 141:104 142:105 143:106 144:10
7 145:108 146:109 147:110 148:110 149:111 150:112 151:112 152:112 153:112 15
4:113 155:114 156:115 157:116 158:117 159:118 160:119 161:120 162:121 163:122
164:122 165:122 166:123 167:123
I0825 16:30:02.140842 140319166060416 run_squad.py:440] token_is_max_context:
11:True 12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:Tr
ue 21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:True 3
0:True 31:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 39:Tru
e 40:True 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:
True 50:True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:True
59:True 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:Tr
ue 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:True 7
8:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 87:Tru
e 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:
True 98:True 99:True 100:True 101:True 102:True 103:True 104:True 105:True 10
6:True 107:True 108:True 109:True 110:True 111:True 112:True 113:True 114:Tru
e 115:True 116:True 117:True 118:True 119:True 120:True 121:True 122:True 12
3:True 124:True 125:True 126:True 127:True 128:True 129:True 130:True 131:Tru
```

```
e 132:True 133:True 134:True 135:True 136:True 137:True 138:True 139:True 14
0:True 141:True 142:True 143:True 144:True 145:True 146:True 147:True 148:Tru
e 149:True 150:True 151:True 152:True 153:True 154:True 155:True 156:True 15
7:True 158:True 159:True 160:True 161:True 162:True 163:True 164:True 165:Tru
e 166:True 167:True
I0825 16:30:02.141901 140319166060416 run_squad.py:442] input_ids: 101 2054 2
001 1996 4323 1997 3565 4605 2753 1029 102 3565 4605 2753 2001 2019 2137 2374
2208 2000 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 2005 1996 23
25 2161 1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169 3249 1996 21
20 2374 3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000 7796 2037 23
53 3565 4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355 1010 2012
11902 1005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254 1010 2662
1012 2004 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 1996 1000 3585
5315 1000 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184 28324 2075
1996 4535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 2015 1006 2104
2029 1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000 1007 1010 20
61 2008 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2753 1012 102 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
I0825 16:30:02.143008 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.144039 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.151964 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.153483 140319166060416 run_squad.py:432] unique_id: 1000000006
I0825 16:30:02.154574 140319166060416 run_squad.py:433] example_index: 6
I0825 16:30:02.155700 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.156881 140319166060416 run_squad.py:436] tokens: [CLS] what da
y was the game played on ? [SEP] super bowl 50 was an american football game
to determine the champion of the national football league ( nfl ) for the 201
5 season . the american football conference ( afc ) champion denver broncos d
efeated the national football conference ( nfc ) champion carolina panthers 2
4 – 10 to earn their third super bowl title . the game was played on february
7 , 2016 , at levi ' s stadium in the san francisco bay area at santa clara ,
california . as this was the 50th super bowl , the league emphasized the " go
lden anniversary " with various gold - themed initiatives , as well as tempor
arily suspend ##ing the tradition of naming each super bowl game with roman n
u ##meral ##s ( under which the game would have been known as " super bowl l
" ) , so that the logo could prominently feature the arabic nu ##meral ##s 50
. [SEP]
I0825 16:30:02.158029 140319166060416 run_squad.py:438] token_to_orig_map: 1
0:0 11:1 12:2 13:3 14:4 15:5 16:6 17:7 18:8 19:9 20:10 21:11 22:12 23:13 24:1
4 25:15 26:16 27:17 28:17 29:17 30:18 31:19 32:20 33:21 34:21 35:22 36:23 37:
24 38:25 39:26 40:26 41:26 42:27 43:28 44:29 45:30 46:31 47:32 48:33 49:34 5
```

```
0:35 51:35 52:35 53:36 54:37 55:38 56:39 57:39 58:39 59:40 60:41 61:42 62:43
63:44 64:45 65:46 66:46 67:47 68:48 69:49 70:50 71:51 72:52 73:53 74:53 75:54
76:54 77:55 78:56 79:56 80:56 81:57 82:58 83:59 84:60 85:61 86:62 87:63 88:64
89:65 90:66 91:66 92:67 93:67 94:68 95:69 96:70 97:71 98:72 99:73 100:74 101:
74 102:75 103:76 104:77 105:78 106:79 107:79 108:80 109:80 110:81 111:82 112:
83 113:83 114:83 115:84 116:84 117:85 118:86 119:87 120:88 121:89 122:89 123:
90 124:91 125:92 126:93 127:94 128:95 129:96 130:97 131:98 132:99 133:100 13
4:100 135:100 136:101 137:101 138:102 139:103 140:104 141:105 142:106 143:107
144:108 145:109 146:110 147:110 148:111 149:112 150:112 151:112 152:112 153:1
13 154:114 155:115 156:116 157:117 158:118 159:119 160:120 161:121 162:122 16
3:122 164:122 165:123 166:123
I0825 16:30:02.159077 140319166060416 run_squad.py:440] token_is_max_context:
10:True 11:True 12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:Tr
ue 20:True 21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:True 2
9:True 30:True 31:True 32:True 33:True 34:True 35:True 36:True 37:True 38:Tru
e 39:True 40:True 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:
True 49:True 50:True 51:True 52:True 53:True 54:True 55:True 56:True 57:True
58:True 59:True 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:Tr
ue 68:True 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:True 7
7:True 78:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True 86:Tru
e 87:True 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:
True 97:True 98:True 99:True 100:True 101:True 102:True 103:True 104:True 10
5:True 106:True 107:True 108:True 109:True 110:True 111:True 112:True 113:Tru
e 114:True 115:True 116:True 117:True 118:True 119:True 120:True 121:True 12
2:True 123:True 124:True 125:True 126:True 127:True 128:True 129:True 130:Tru
e 131:True 132:True 133:True 134:True 135:True 136:True 137:True 138:True 13
9:True 140:True 141:True 142:True 143:True 144:True 145:True 146:True 147:Tru
e 148:True 149:True 150:True 151:True 152:True 153:True 154:True 155:True 15
6:True 157:True 158:True 159:True 160:True 161:True 162:True 163:True 164:Tru
e 165:True 166:True
I0825 16:30:02.160294 140319166060416 run_squad.py:442] input_ids: 101 2054 2
154 2001 1996 2208 2209 2006 1029 102 3565 4605 2753 2001 2019 2137 2374 2208
2000 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 2005 1996 2325 21
61 1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169 3249 1996 2120 23
74 3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000 7796 2037 2353 35
65 4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355 1010 2012 1190
2 1005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254 1010 2662 1012
2004 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 1996 1000 3585 5315
1000 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184 28324 2075 1996
4535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 2015 1006 2104 2029
1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000 1007 1010 2061 20
08 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2753 1012 102 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0
I0825 16:30:02.161386 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.162557 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.170413 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.171831 140319166060416 run_squad.py:432] unique_id: 1000000007
I0825 16:30:02.173029 140319166060416 run_squad.py:433] example_index: 7
I0825 16:30:02.174124 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.175318 140319166060416 run_squad.py:436] tokens: [CLS] what is
the afc short for ? [SEP] super bowl 50 was an american football game to dete
rmine the champion of the national football league ( nfl ) for the 2015 seaso
n . the american football conference ( afc ) champion denver broncos defeated
the national football conference ( nfc ) champion carolina panthers 24 – 10 t
o earn their third super bowl title . the game was played on february 7 , 201
6 , at levi ' s stadium in the san francisco bay area at santa clara , califo
rnia . as this was the 50th super bowl , the league emphasized the " golden a
nniversary " with various gold - themed initiatives , as well as temporarily
suspend ##ing the tradition of naming each super bowl game with roman nu ##me
ral ##s ( under which the game would have been known as " super bowl l " ) ,
so that the logo could prominently feature the arabic nu ##meral ##s 50 . [SE
P]
I0825 16:30:02.176428 140319166060416 run_squad.py:438] token_to_orig_map: 9:
0 10:1 11:2 12:3 13:4 14:5 15:6 16:7 17:8 18:9 19:10 20:11 21:12 22:13 23:14
24:15 25:16 26:17 27:17 28:17 29:18 30:19 31:20 32:21 33:21 34:22 35:23 36:24
37:25 38:26 39:26 40:26 41:27 42:28 43:29 44:30 45:31 46:32 47:33 48:34 49:35
50:35 51:35 52:36 53:37 54:38 55:39 56:39 57:39 58:40 59:41 60:42 61:43 62:44
63:45 64:46 65:46 66:47 67:48 68:49 69:50 70:51 71:52 72:53 73:53 74:54 75:54
76:55 77:56 78:56 79:56 80:57 81:58 82:59 83:60 84:61 85:62 86:63 87:64 88:65
89:66 90:66 91:67 92:67 93:68 94:69 95:70 96:71 97:72 98:73 99:74 100:74 101:
75 102:76 103:77 104:78 105:79 106:79 107:80 108:80 109:81 110:82 111:83 112:
83 113:83 114:84 115:84 116:85 117:86 118:87 119:88 120:89 121:89 122:90 123:
91 124:92 125:93 126:94 127:95 128:96 129:97 130:98 131:99 132:100 133:100 13
4:100 135:101 136:101 137:102 138:103 139:104 140:105 141:106 142:107 143:108
144:109 145:110 146:110 147:111 148:112 149:112 150:112 151:112 152:113 153:1
14 154:115 155:116 156:117 157:118 158:119 159:120 160:121 161:122 162:122 16
3:122 164:123 165:123
I0825 16:30:02.177598 140319166060416 run_squad.py:440] token_is_max_context:
9:True 10:True 11:True 12:True 13:True 14:True 15:True 16:True 17:True 18:Tru
e 19:True 20:True 21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:
True 29:True 30:True 31:True 32:True 33:True 34:True 35:True 36:True 37:True
38:True 39:True 40:True 41:True 42:True 43:True 44:True 45:True 46:True 47:Tr
ue 48:True 49:True 50:True 51:True 52:True 53:True 54:True 55:True 56:True 5
7:True 58:True 59:True 60:True 61:True 62:True 63:True 64:True 65:True 66:Tru
e 67:True 68:True 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:
True 77:True 78:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True
86:True 87:True 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:Tr
ue 96:True 97:True 98:True 99:True 100:True 101:True 102:True 103:True 104:Tr
ue 105:True 106:True 107:True 108:True 109:True 110:True 111:True 112:True 11
3:True 114:True 115:True 116:True 117:True 118:True 119:True 120:True 121:Tru
e 122:True 123:True 124:True 125:True 126:True 127:True 128:True 129:True 13
0:True 131:True 132:True 133:True 134:True 135:True 136:True 137:True 138:Tru
e 139:True 140:True 141:True 142:True 143:True 144:True 145:True 146:True 14
7:True 148:True 149:True 150:True 151:True 152:True 153:True 154:True 155:Tru
e 156:True 157:True 158:True 159:True 160:True 161:True 162:True 163:True 16
4:True 165:True
I0825 16:30:02.178793 140319166060416 run_squad.py:442] input_ids: 101 2054 2
003 1996 10511 2460 2005 1029 102 3565 4605 2753 2001 2019 2137 2374 2208 200
```

```
0 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 2005 1996 2325 2161
1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169 3249 1996 2120 2374
3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000 7796 2037 2353 3565
4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355 1010 2012 11902 1
005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254 1010 2662 1012 20
04 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 1996 1000 3585 5315 10
00 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184 28324 2075 1996 4
535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 2015 1006 2104 2029
1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000 1007 1010 2061 20
08 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2753 1012 102 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
I0825 16:30:02.179903 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.180976 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.189033 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.190600 140319166060416 run_squad.py:432] unique_id: 1000000008
I0825 16:30:02.191840 140319166060416 run_squad.py:433] example_index: 8
I0825 16:30:02.192935 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.194063 140319166060416 run_squad.py:436] tokens: [CLS] what wa
s the theme of super bowl 50 ? [SEP] super bowl 50 was an american football g
ame to determine the champion of the national football league ( nfl ) for the
2015 season . the american football conference ( afc ) champion denver bronco
s defeated the national football conference ( nfc ) champion carolina panther
s 24 – 10 to earn their third super bowl title . the game was played on febru
ary 7 , 2016 , at levi ' s stadium in the san francisco bay area at santa cla
ra , california . as this was the 50th super bowl , the league emphasized the
" golden anniversary " with various gold - themed initiatives , as well as te
mporarily suspend ##ing the tradition of naming each super bowl game with rom
an nu ##meral ##s ( under which the game would have been known as " super bow
l l " ) , so that the logo could prominently feature the arabic nu ##meral ##
s 50 . [SEP]
I0825 16:30:02.200030 140319166060416 run_squad.py:438] token_to_orig_map: 1
1:0 12:1 13:2 14:3 15:4 16:5 17:6 18:7 19:8 20:9 21:10 22:11 23:12 24:13 25:1
4 26:15 27:16 28:17 29:17 30:17 31:18 32:19 33:20 34:21 35:21 36:22 37:23 38:
24 39:25 40:26 41:26 42:26 43:27 44:28 45:29 46:30 47:31 48:32 49:33 50:34 5
1:35 52:35 53:35 54:36 55:37 56:38 57:39 58:39 59:39 60:40 61:41 62:42 63:43
 64:44 65:45 66:46 67:46 68:47 69:48 70:49 71:50 72:51 73:52 74:53 75:53 76:5
4 77:54 78:55 79:56 80:56 81:56 82:57 83:58 84:59 85:60 86:61 87:62 88:63 89:
64 90:65 91:66 92:66 93:67 94:67 95:68 96:69 97:70 98:71 99:72 100:73 101:74
 102:74 103:75 104:76 105:77 106:78 107:79 108:79 109:80 110:80 111:81 112:82
113:83 114:83 115:83 116:84 117:84 118:85 119:86 120:87 121:88 122:89 123:89
 124:90 125:91 126:92 127:93 128:94 129:95 130:96 131:97 132:98 133:99 134:10
```

```
0 135:100 136:100 137:101 138:101 139:102 140:103 141:104 142:105 143:106 14
4:107 145:108 146:109 147:110 148:110 149:111 150:112 151:112 152:112 153:112
154:113 155:114 156:115 157:116 158:117 159:118 160:119 161:120 162:121 163:1
22 164:122 165:122 166:123 167:123
I0825 16:30:02.201228 140319166060416 run_squad.py:440] token_is_max_context:
11:True 12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:Tr
ue 21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:True 3
0:True 31:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 39:Tru
e 40:True 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:
True 50:True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:True
 59:True 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:T
rue 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:True 7
8:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 87:Tru
e 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:
True 98:True 99:True 100:True 101:True 102:True 103:True 104:True 105:True 10
6:True 107:True 108:True 109:True 110:True 111:True 112:True 113:True 114:Tru
e 115:True 116:True 117:True 118:True 119:True 120:True 121:True 122:True 12
3:True 124:True 125:True 126:True 127:True 128:True 129:True 130:True 131:Tru
e 132:True 133:True 134:True 135:True 136:True 137:True 138:True 139:True 14
0:True 141:True 142:True 143:True 144:True 145:True 146:True 147:True 148:Tru
e 149:True 150:True 151:True 152:True 153:True 154:True 155:True 156:True 15
7:True 158:True 159:True 160:True 161:True 162:True 163:True 164:True 165:Tru
e 166:True 167:True
I0825 16:30:02.202337 140319166060416 run_squad.py:442] input_ids: 101 2054 2
001 1996 4323 1997 3565 4605 2753 1029 102 3565 4605 2753 2001 2019 2137 2374
2208 2000 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 2005 1996 23
25 2161 1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169 3249 1996 21
20 2374 3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000 7796 2037 23
53 3565 4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355 1010 2012
11902 1005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254 1010 2662
 1012 2004 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 1996 1000 3585
5315 1000 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184 28324 2075
1996 4535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 2015 1006 2104
2029 1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000 1007 1010 20
61 2008 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2753 1012 102 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.203266 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.204290 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.211796 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.213268 140319166060416 run_squad.py:432] unique_id: 1000000009
I0825 16:30:02.214401 140319166060416 run_squad.py:433] example_index: 9
```

```
I0825 16:30:02.215502 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.216636 140319166060416 run_squad.py:436] tokens: [CLS] what do
es afc stand for ? [SEP] super bowl 50 was an american football game to deter
mine the champion of the national football league ( nfl ) for the 2015 season
. the american football conference ( afc ) champion denver broncos defeated t
he national football conference ( nfc ) champion carolina panthers 24 – 10 to
earn their third super bowl title . the game was played on february 7 , 2016
 , at levi ' s stadium in the san francisco bay area at santa clara , califor
nia . as this was the 50th super bowl , the league emphasized the " golden an
niversary " with various gold - themed initiatives , as well as temporarily s
uspend ##ing the tradition of naming each super bowl game with roman nu ##mer
al ##s ( under which the game would have been known as " super bowl l " ) , s
o that the logo could prominently feature the arabic nu ##meral ##s 50 . [SE
P]
I0825 16:30:02.217759 140319166060416 run_squad.py:438] token_to_orig_map: 8:
0 9:1 10:2 11:3 12:4 13:5 14:6 15:7 16:8 17:9 18:10 19:11 20:12 21:13 22:14 2
3:15 24:16 25:17 26:17 27:17 28:18 29:19 30:20 31:21 32:21 33:22 34:23 35:24
 36:25 37:26 38:26 39:26 40:27 41:28 42:29 43:30 44:31 45:32 46:33 47:34 48:3
5 49:35 50:35 51:36 52:37 53:38 54:39 55:39 56:39 57:40 58:41 59:42 60:43 61:
44 62:45 63:46 64:46 65:47 66:48 67:49 68:50 69:51 70:52 71:53 72:53 73:54 7
4:54 75:55 76:56 77:56 78:56 79:57 80:58 81:59 82:60 83:61 84:62 85:63 86:64
 87:65 88:66 89:66 90:67 91:67 92:68 93:69 94:70 95:71 96:72 97:73 98:74 99:7
4 100:75 101:76 102:77 103:78 104:79 105:79 106:80 107:80 108:81 109:82 110:8
3 111:83 112:83 113:84 114:84 115:85 116:86 117:87 118:88 119:89 120:89 121:9
0 122:91 123:92 124:93 125:94 126:95 127:96 128:97 129:98 130:99 131:100 132:
100 133:100 134:101 135:101 136:102 137:103 138:104 139:105 140:106 141:107 1
42:108 143:109 144:110 145:110 146:111 147:112 148:112 149:112 150:112 151:11
3 152:114 153:115 154:116 155:117 156:118 157:119 158:120 159:121 160:122 16
1:122 162:122 163:123 164:123
I0825 16:30:02.218853 140319166060416 run_squad.py:440] token_is_max_context:
8:True 9:True 10:True 11:True 12:True 13:True 14:True 15:True 16:True 17:True
18:True 19:True 20:True 21:True 22:True 23:True 24:True 25:True 26:True 27:Tr
ue 28:True 29:True 30:True 31:True 32:True 33:True 34:True 35:True 36:True 3
7:True 38:True 39:True 40:True 41:True 42:True 43:True 44:True 45:True 46:Tru
e 47:True 48:True 49:True 50:True 51:True 52:True 53:True 54:True 55:True 56:
True 57:True 58:True 59:True 60:True 61:True 62:True 63:True 64:True 65:True
 66:True 67:True 68:True 69:True 70:True 71:True 72:True 73:True 74:True 75:T
rue 76:True 77:True 78:True 79:True 80:True 81:True 82:True 83:True 84:True 8
5:True 86:True 87:True 88:True 89:True 90:True 91:True 92:True 93:True 94:Tru
e 95:True 96:True 97:True 98:True 99:True 100:True 101:True 102:True 103:True
104:True 105:True 106:True 107:True 108:True 109:True 110:True 111:True 112:T
rue 113:True 114:True 115:True 116:True 117:True 118:True 119:True 120:True 1
21:True 122:True 123:True 124:True 125:True 126:True 127:True 128:True 129:Tr
ue 130:True 131:True 132:True 133:True 134:True 135:True 136:True 137:True 13
8:True 139:True 140:True 141:True 142:True 143:True 144:True 145:True 146:Tru
e 147:True 148:True 149:True 150:True 151:True 152:True 153:True 154:True 15
5:True 156:True 157:True 158:True 159:True 160:True 161:True 162:True 163:Tru
e 164:True
I0825 16:30:02.219900 140319166060416 run_squad.py:442] input_ids: 101 2054 2
515 10511 3233 2005 1029 102 3565 4605 2753 2001 2019 2137 2374 2208 2000 564
6 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 2005 1996 2325 2161 1012
 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169 3249 1996 2120 2374 3034
1006 22309 1007 3410 3792 12915 2484 1516 2184 2000 7796 2037 2353 3565 4605
 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355 1010 2012 11902 1005
 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254 1010 2662 1012 2004
 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 1996 1000 3585 5315 1000
2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184 28324 2075 1996 4535
```

```
1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 2015 1006 2104 2029 1996
2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000 1007 1010 2061 2008 19
96 8154 2071 14500 3444 1996 5640 16371 28990 2015 2753 1012 102 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0
I0825 16:30:02.220910 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.221891 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.229490 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.231020 140319166060416 run_squad.py:432] unique_id: 1000000010
I0825 16:30:02.232181 140319166060416 run_squad.py:433] example_index: 10
I0825 16:30:02.233230 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.234359 140319166060416 run_squad.py:436] tokens: [CLS] what da
y was the super bowl played on ? [SEP] super bowl 50 was an american football
game to determine the champion of the national football league ( nfl ) for th
e 2015 season . the american football conference ( afc ) champion denver bron
cos defeated the national football conference ( nfc ) champion carolina panth
ers 24 – 10 to earn their third super bowl title . the game was played on feb
ruary 7 , 2016 , at levi ' s stadium in the san francisco bay area at santa c
lara , california . as this was the 50th super bowl , the league emphasized t
he " golden anniversary " with various gold - themed initiatives , as well as
temporarily suspend ##ing the tradition of naming each super bowl game with r
oman nu ##meral ##s ( under which the game would have been known as " super b
owl l " ) , so that the logo could prominently feature the arabic nu ##meral
 ##s 50 . [SEP]
I0825 16:30:02.235504 140319166060416 run_squad.py:438] token_to_orig_map: 1
1:0 12:1 13:2 14:3 15:4 16:5 17:6 18:7 19:8 20:9 21:10 22:11 23:12 24:13 25:1
4 26:15 27:16 28:17 29:17 30:17 31:18 32:19 33:20 34:21 35:21 36:22 37:23 38:
24 39:25 40:26 41:26 42:26 43:27 44:28 45:29 46:30 47:31 48:32 49:33 50:34 5
1:35 52:35 53:35 54:36 55:37 56:38 57:39 58:39 59:39 60:40 61:41 62:42 63:43
 64:44 65:45 66:46 67:46 68:47 69:48 70:49 71:50 72:51 73:52 74:53 75:53 76:5
4 77:54 78:55 79:56 80:56 81:56 82:57 83:58 84:59 85:60 86:61 87:62 88:63 89:
64 90:65 91:66 92:66 93:67 94:67 95:68 96:69 97:70 98:71 99:72 100:73 101:74
 102:74 103:75 104:76 105:77 106:78 107:79 108:79 109:80 110:80 111:81 112:82
113:83 114:83 115:83 116:84 117:84 118:85 119:86 120:87 121:88 122:89 123:89
 124:90 125:91 126:92 127:93 128:94 129:95 130:96 131:97 132:98 133:99 134:10
0 135:100 136:100 137:101 138:101 139:102 140:103 141:104 142:105 143:106 14
4:107 145:108 146:109 147:110 148:110 149:111 150:112 151:112 152:112 153:112
154:113 155:114 156:115 157:116 158:117 159:118 160:119 161:120 162:121 163:1
22 164:122 165:122 166:123 167:123
I0825 16:30:02.236648 140319166060416 run_squad.py:440] token_is_max_context:
11:True 12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:Tr
ue 21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:True 3
```

```
0:True 31:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 39:Tru
e 40:True 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:
True 50:True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:True
 59:True 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:T
rue 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:True 7
8:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 87:Tru
e 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:
True 98:True 99:True 100:True 101:True 102:True 103:True 104:True 105:True 10
6:True 107:True 108:True 109:True 110:True 111:True 112:True 113:True 114:Tru
e 115:True 116:True 117:True 118:True 119:True 120:True 121:True 122:True 12
3:True 124:True 125:True 126:True 127:True 128:True 129:True 130:True 131:Tru
e 132:True 133:True 134:True 135:True 136:True 137:True 138:True 139:True 14
0:True 141:True 142:True 143:True 144:True 145:True 146:True 147:True 148:Tru
e 149:True 150:True 151:True 152:True 153:True 154:True 155:True 156:True 15
7:True 158:True 159:True 160:True 161:True 162:True 163:True 164:True 165:Tru
e 166:True 167:True
I0825 16:30:02.237740 140319166060416 run_squad.py:442] input_ids: 101 2054 2
154 2001 1996 3565 4605 2209 2006 1029 102 3565 4605 2753 2001 2019 2137 2374
2208 2000 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 2005 1996 23
25 2161 1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169 3249 1996 21
20 2374 3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000 7796 2037 23
53 3565 4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355 1010 2012
11902 1005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254 1010 2662
 1012 2004 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 1996 1000 3585
5315 1000 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184 28324 2075
1996 4535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 2015 1006 2104
2029 1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000 1007 1010 20
61 2008 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2753 1012 102 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.238890 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.239958 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.247958 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.249489 140319166060416 run_squad.py:432] unique_id: 1000000011
I0825 16:30:02.250677 140319166060416 run_squad.py:433] example_index: 11
I0825 16:30:02.251824 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.253088 140319166060416 run_squad.py:436] tokens: [CLS] who won
super bowl 50 ? [SEP] super bowl 50 was an american football game to determin
e the champion of the national football league ( nfl ) for the 2015 season .
 the american football conference ( afc ) champion denver broncos defeated th
e national football conference ( nfc ) champion carolina panthers 24 – 10 to
 earn their third super bowl title . the game was played on february 7 , 2016
```

```
, at levi ' s stadium in the san francisco bay area at santa clara , californ
ia . as this was the 50th super bowl , the league emphasized the " golden ann
iversary " with various gold - themed initiatives , as well as temporarily su
spend ##ing the tradition of naming each super bowl game with roman nu ##mera
l ##s ( under which the game would have been known as " super bowl l " ) , so
that the logo could prominently feature the arabic nu ##meral ##s 50 . [SEP]
I0825 16:30:02.254157 140319166060416 run_squad.py:438] token_to_orig_map: 8:
0 9:1 10:2 11:3 12:4 13:5 14:6 15:7 16:8 17:9 18:10 19:11 20:12 21:13 22:14 2
3:15 24:16 25:17 26:17 27:17 28:18 29:19 30:20 31:21 32:21 33:22 34:23 35:24
 36:25 37:26 38:26 39:26 40:27 41:28 42:29 43:30 44:31 45:32 46:33 47:34 48:3
5 49:35 50:35 51:36 52:37 53:38 54:39 55:39 56:39 57:40 58:41 59:42 60:43 61:
44 62:45 63:46 64:46 65:47 66:48 67:49 68:50 69:51 70:52 71:53 72:53 73:54 7
4:54 75:55 76:56 77:56 78:56 79:57 80:58 81:59 82:60 83:61 84:62 85:63 86:64
 87:65 88:66 89:66 90:67 91:67 92:68 93:69 94:70 95:71 96:72 97:73 98:74 99:7
4 100:75 101:76 102:77 103:78 104:79 105:79 106:80 107:80 108:81 109:82 110:8
3 111:83 112:83 113:84 114:84 115:85 116:86 117:87 118:88 119:89 120:89 121:9
0 122:91 123:92 124:93 125:94 126:95 127:96 128:97 129:98 130:99 131:100 132:
100 133:100 134:101 135:101 136:102 137:103 138:104 139:105 140:106 141:107 1
42:108 143:109 144:110 145:110 146:111 147:112 148:112 149:112 150:112 151:11
3 152:114 153:115 154:116 155:117 156:118 157:119 158:120 159:121 160:122 16
1:122 162:122 163:123 164:123
I0825 16:30:02.255315 140319166060416 run_squad.py:440] token_is_max_context:
8:True 9:True 10:True 11:True 12:True 13:True 14:True 15:True 16:True 17:True
18:True 19:True 20:True 21:True 22:True 23:True 24:True 25:True 26:True 27:Tr
ue 28:True 29:True 30:True 31:True 32:True 33:True 34:True 35:True 36:True 3
7:True 38:True 39:True 40:True 41:True 42:True 43:True 44:True 45:True 46:Tru
e 47:True 48:True 49:True 50:True 51:True 52:True 53:True 54:True 55:True 56:
True 57:True 58:True 59:True 60:True 61:True 62:True 63:True 64:True 65:True
 66:True 67:True 68:True 69:True 70:True 71:True 72:True 73:True 74:True 75:T
rue 76:True 77:True 78:True 79:True 80:True 81:True 82:True 83:True 84:True 8
5:True 86:True 87:True 88:True 89:True 90:True 91:True 92:True 93:True 94:Tru
e 95:True 96:True 97:True 98:True 99:True 100:True 101:True 102:True 103:True
104:True 105:True 106:True 107:True 108:True 109:True 110:True 111:True 112:T
rue 113:True 114:True 115:True 116:True 117:True 118:True 119:True 120:True 1
21:True 122:True 123:True 124:True 125:True 126:True 127:True 128:True 129:Tr
ue 130:True 131:True 132:True 133:True 134:True 135:True 136:True 137:True 13
8:True 139:True 140:True 141:True 142:True 143:True 144:True 145:True 146:Tru
e 147:True 148:True 149:True 150:True 151:True 152:True 153:True 154:True 15
5:True 156:True 157:True 158:True 159:True 160:True 161:True 162:True 163:Tru
e 164:True
I0825 16:30:02.256422 140319166060416 run_squad.py:442] input_ids: 101 2040 2
180 3565 4605 2753 1029 102 3565 4605 2753 2001 2019 2137 2374 2208 2000 5646
1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 2005 1996 2325 2161 1012 19
96 2137 2374 3034 1006 10511 1007 3410 7573 14169 3249 1996 2120 2374 3034 10
06 22309 1007 3410 3792 12915 2484 1516 2184 2000 7796 2037 2353 3565 4605 25
16 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355 1010 2012 11902 1005 105
5 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254 1010 2662 1012 2004 2023
2001 1996 12951 3565 4605 1010 1996 2223 13155 1996 1000 3585 5315 1000 2007
 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184 28324 2075 1996 4535 199
7 10324 2169 3565 4605 2208 2007 3142 16371 28990 2015 1006 2104 2029 1996 22
08 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000 1007 1010 2061 2008 1996
8154 2071 14500 3444 1996 5640 16371 28990 2015 2753 1012 102 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
I0825 16:30:02.257555 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.258620 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.267003 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.268903 140319166060416 run_squad.py:432] unique_id: 1000000012
I0825 16:30:02.269942 140319166060416 run_squad.py:433] example_index: 12
I0825 16:30:02.270941 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.272073 140319166060416 run_squad.py:436] tokens: [CLS] what ve
nue did super bowl 50 take place in ? [SEP] super bowl 50 was an american foo
tball game to determine the champion of the national football league ( nfl )
 for the 2015 season . the american football conference ( afc ) champion denv
er broncos defeated the national football conference ( nfc ) champion carolin
a panthers 24 – 10 to earn their third super bowl title . the game was played
on february 7 , 2016 , at levi ' s stadium in the san francisco bay area at s
anta clara , california . as this was the 50th super bowl , the league emphas
ized the " golden anniversary " with various gold - themed initiatives , as w
ell as temporarily suspend ##ing the tradition of naming each super bowl game
with roman nu ##meral ##s ( under which the game would have been known as " s
uper bowl l " ) , so that the logo could prominently feature the arabic nu ##
meral ##s 50 . [SEP]
I0825 16:30:02.273071 140319166060416 run_squad.py:438] token_to_orig_map: 1
2:0 13:1 14:2 15:3 16:4 17:5 18:6 19:7 20:8 21:9 22:10 23:11 24:12 25:13 26:1
4 27:15 28:16 29:17 30:17 31:17 32:18 33:19 34:20 35:21 36:21 37:22 38:23 39:
24 40:25 41:26 42:26 43:26 44:27 45:28 46:29 47:30 48:31 49:32 50:33 51:34 5
2:35 53:35 54:35 55:36 56:37 57:38 58:39 59:39 60:39 61:40 62:41 63:42 64:43
 65:44 66:45 67:46 68:46 69:47 70:48 71:49 72:50 73:51 74:52 75:53 76:53 77:5
4 78:54 79:55 80:56 81:56 82:56 83:57 84:58 85:59 86:60 87:61 88:62 89:63 90:
64 91:65 92:66 93:66 94:67 95:67 96:68 97:69 98:70 99:71 100:72 101:73 102:74
103:74 104:75 105:76 106:77 107:78 108:79 109:79 110:80 111:80 112:81 113:82
 114:83 115:83 116:83 117:84 118:84 119:85 120:86 121:87 122:88 123:89 124:89
125:90 126:91 127:92 128:93 129:94 130:95 131:96 132:97 133:98 134:99 135:100
136:100 137:100 138:101 139:101 140:102 141:103 142:104 143:105 144:106 145:1
07 146:108 147:109 148:110 149:110 150:111 151:112 152:112 153:112 154:112 15
5:113 156:114 157:115 158:116 159:117 160:118 161:119 162:120 163:121 164:122
165:122 166:122 167:123 168:123
I0825 16:30:02.274098 140319166060416 run_squad.py:440] token_is_max_context:
12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:Tr
ue 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:True 3
1:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 40:Tru
e 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:
True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:True
 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:T
rue 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:True 7
9:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 88:Tru
e 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:
True 99:True 100:True 101:True 102:True 103:True 104:True 105:True 106:True 1
```

```
07:True 108:True 109:True 110:True 111:True 112:True 113:True 114:True 115:Tr
ue 116:True 117:True 118:True 119:True 120:True 121:True 122:True 123:True 12
4:True 125:True 126:True 127:True 128:True 129:True 130:True 131:True 132:Tru
e 133:True 134:True 135:True 136:True 137:True 138:True 139:True 140:True 14
1:True 142:True 143:True 144:True 145:True 146:True 147:True 148:True 149:Tru
e 150:True 151:True 152:True 153:True 154:True 155:True 156:True 157:True 15
8:True 159:True 160:True 161:True 162:True 163:True 164:True 165:True 166:Tru
e 167:True 168:True
I0825 16:30:02.275074 140319166060416 run_squad.py:442] input_ids: 101 2054 6
891 2106 3565 4605 2753 2202 2173 1999 1029 102 3565 4605 2753 2001 2019 2137
2374 2208 2000 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 2005 19
96 2325 2161 1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169 3249 19
96 2120 2374 3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000 7796 20
37 2353 3565 4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355 1010
2012 11902 1005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254 1010
 2662 1012 2004 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 1996 1000
3585 5315 1000 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184 28324
2075 1996 4535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 2015 1006
2104 2029 1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000 1007 10
10 2061 2008 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2753 1012 1
02 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.276067 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.277087 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.285183 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.287172 140319166060416 run_squad.py:432] unique_id: 1000000013
I0825 16:30:02.288293 140319166060416 run_squad.py:433] example_index: 13
I0825 16:30:02.289354 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.290478 140319166060416 run_squad.py:436] tokens: [CLS] what ci
ty did super bowl 50 take place in ? [SEP] super bowl 50 was an american foot
ball game to determine the champion of the national football league ( nfl ) f
or the 2015 season . the american football conference ( afc ) champion denver
broncos defeated the national football conference ( nfc ) champion carolina p
anthers 24 – 10 to earn their third super bowl title . the game was played on
february 7 , 2016 , at levi ' s stadium in the san francisco bay area at sant
a clara , california . as this was the 50th super bowl , the league emphasize
d the " golden anniversary " with various gold - themed initiatives , as well
as temporarily suspend ##ing the tradition of naming each super bowl game wit
h roman nu ##meral ##s ( under which the game would have been known as " supe
r bowl l " ) , so that the logo could prominently feature the arabic nu ##mer
al ##s 50 . [SEP]
I0825 16:30:02.291667 140319166060416 run_squad.py:438] token_to_orig_map: 1
```

```
2:0 13:1 14:2 15:3 16:4 17:5 18:6 19:7 20:8 21:9 22:10 23:11 24:12 25:13 26:1
4 27:15 28:16 29:17 30:17 31:17 32:18 33:19 34:20 35:21 36:21 37:22 38:23 39:
24 40:25 41:26 42:26 43:26 44:27 45:28 46:29 47:30 48:31 49:32 50:33 51:34 5
2:35 53:35 54:35 55:36 56:37 57:38 58:39 59:39 60:39 61:40 62:41 63:42 64:43
 65:44 66:45 67:46 68:46 69:47 70:48 71:49 72:50 73:51 74:52 75:53 76:53 77:5
4 78:54 79:55 80:56 81:56 82:56 83:57 84:58 85:59 86:60 87:61 88:62 89:63 90:
64 91:65 92:66 93:66 94:67 95:67 96:68 97:69 98:70 99:71 100:72 101:73 102:74
103:74 104:75 105:76 106:77 107:78 108:79 109:79 110:80 111:80 112:81 113:82
 114:83 115:83 116:83 117:84 118:84 119:85 120:86 121:87 122:88 123:89 124:89
125:90 126:91 127:92 128:93 129:94 130:95 131:96 132:97 133:98 134:99 135:100
136:100 137:100 138:101 139:101 140:102 141:103 142:104 143:105 144:106 145:1
07 146:108 147:109 148:110 149:110 150:111 151:112 152:112 153:112 154:112 15
5:113 156:114 157:115 158:116 159:117 160:118 161:119 162:120 163:121 164:122
165:122 166:122 167:123 168:123
I0825 16:30:02.292848 140319166060416 run_squad.py:440] token_is_max_context:
12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:Tr
ue 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:True 3
1:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 40:Tru
e 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:
True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:True
 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:T
rue 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:True 7
9:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 88:Tru
e 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:
True 99:True 100:True 101:True 102:True 103:True 104:True 105:True 106:True 1
07:True 108:True 109:True 110:True 111:True 112:True 113:True 114:True 115:Tr
ue 116:True 117:True 118:True 119:True 120:True 121:True 122:True 123:True 12
4:True 125:True 126:True 127:True 128:True 129:True 130:True 131:True 132:Tru
e 133:True 134:True 135:True 136:True 137:True 138:True 139:True 140:True 14
1:True 142:True 143:True 144:True 145:True 146:True 147:True 148:True 149:Tru
e 150:True 151:True 152:True 153:True 154:True 155:True 156:True 157:True 15
8:True 159:True 160:True 161:True 162:True 163:True 164:True 165:True 166:Tru
e 167:True 168:True
I0825 16:30:02.293986 140319166060416 run_squad.py:442] input_ids: 101 2054 2
103 2106 3565 4605 2753 2202 2173 1999 1029 102 3565 4605 2753 2001 2019 2137
2374 2208 2000 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 2005 19
96 2325 2161 1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169 3249 19
96 2120 2374 3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000 7796 20
37 2353 3565 4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355 1010
2012 11902 1005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254 1010
 2662 1012 2004 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 1996 1000
3585 5315 1000 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184 28324
2075 1996 4535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 2015 1006
2104 2029 1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000 1007 10
10 2061 2008 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2753 1012 1
02 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.295094 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.296200 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
```

```
 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.304545 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.305963 140319166060416 run_squad.py:432] unique_id: 1000000014
I0825 16:30:02.307604 140319166060416 run_squad.py:433] example_index: 14
I0825 16:30:02.308620 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.309709 140319166060416 run_squad.py:436] tokens: [CLS] if roma
n nu ##meral ##s were used , what would super bowl 50 have been called ? [SE
P] super bowl 50 was an american football game to determine the champion of t
he national football league ( nfl ) for the 2015 season . the american footba
ll conference ( afc ) champion denver broncos defeated the national football
 conference ( nfc ) champion carolina panthers 24 – 10 to earn their third su
per bowl title . the game was played on february 7 , 2016 , at levi ' s stadi
um in the san francisco bay area at santa clara , california . as this was th
e 50th super bowl , the league emphasized the " golden anniversary " with var
ious gold - themed initiatives , as well as temporarily suspend ##ing the tra
dition of naming each super bowl game with roman nu ##meral ##s ( under which
the game would have been known as " super bowl l " ) , so that the logo could
prominently feature the arabic nu ##meral ##s 50 . [SEP]
I0825 16:30:02.310820 140319166060416 run_squad.py:438] token_to_orig_map: 1
9:0 20:1 21:2 22:3 23:4 24:5 25:6 26:7 27:8 28:9 29:10 30:11 31:12 32:13 33:1
4 34:15 35:16 36:17 37:17 38:17 39:18 40:19 41:20 42:21 43:21 44:22 45:23 46:
24 47:25 48:26 49:26 50:26 51:27 52:28 53:29 54:30 55:31 56:32 57:33 58:34 5
9:35 60:35 61:35 62:36 63:37 64:38 65:39 66:39 67:39 68:40 69:41 70:42 71:43
 72:44 73:45 74:46 75:46 76:47 77:48 78:49 79:50 80:51 81:52 82:53 83:53 84:5
4 85:54 86:55 87:56 88:56 89:56 90:57 91:58 92:59 93:60 94:61 95:62 96:63 97:
64 98:65 99:66 100:66 101:67 102:67 103:68 104:69 105:70 106:71 107:72 108:73
109:74 110:74 111:75 112:76 113:77 114:78 115:79 116:79 117:80 118:80 119:81
 120:82 121:83 122:83 123:83 124:84 125:84 126:85 127:86 128:87 129:88 130:89
131:89 132:90 133:91 134:92 135:93 136:94 137:95 138:96 139:97 140:98 141:99
 142:100 143:100 144:100 145:101 146:101 147:102 148:103 149:104 150:105 151:
106 152:107 153:108 154:109 155:110 156:110 157:111 158:112 159:112 160:112 1
61:112 162:113 163:114 164:115 165:116 166:117 167:118 168:119 169:120 170:12
1 171:122 172:122 173:122 174:123 175:123
I0825 16:30:02.311895 140319166060416 run_squad.py:440] token_is_max_context:
19:True 20:True 21:True 22:True 23:True 24:True 25:True 26:True 27:True 28:Tr
ue 29:True 30:True 31:True 32:True 33:True 34:True 35:True 36:True 37:True 3
8:True 39:True 40:True 41:True 42:True 43:True 44:True 45:True 46:True 47:Tru
e 48:True 49:True 50:True 51:True 52:True 53:True 54:True 55:True 56:True 57:
True 58:True 59:True 60:True 61:True 62:True 63:True 64:True 65:True 66:True
 67:True 68:True 69:True 70:True 71:True 72:True 73:True 74:True 75:True 76:T
rue 77:True 78:True 79:True 80:True 81:True 82:True 83:True 84:True 85:True 8
6:True 87:True 88:True 89:True 90:True 91:True 92:True 93:True 94:True 95:Tru
e 96:True 97:True 98:True 99:True 100:True 101:True 102:True 103:True 104:Tru
e 105:True 106:True 107:True 108:True 109:True 110:True 111:True 112:True 11
3:True 114:True 115:True 116:True 117:True 118:True 119:True 120:True 121:Tru
e 122:True 123:True 124:True 125:True 126:True 127:True 128:True 129:True 13
0:True 131:True 132:True 133:True 134:True 135:True 136:True 137:True 138:Tru
e 139:True 140:True 141:True 142:True 143:True 144:True 145:True 146:True 14
7:True 148:True 149:True 150:True 151:True 152:True 153:True 154:True 155:Tru
e 156:True 157:True 158:True 159:True 160:True 161:True 162:True 163:True 16
4:True 165:True 166:True 167:True 168:True 169:True 170:True 171:True 172:Tru
```

```
e 173:True 174:True 175:True
I0825 16:30:02.313009 140319166060416 run_squad.py:442] input_ids: 101 2065 3
142 16371 28990 2015 2020 2109 1010 2054 2052 3565 4605 2753 2031 2042 2170 1
029 102 3565 4605 2753 2001 2019 2137 2374 2208 2000 5646 1996 3410 1997 1996
2120 2374 2223 1006 5088 1007 2005 1996 2325 2161 1012 1996 2137 2374 3034 10
06 10511 1007 3410 7573 14169 3249 1996 2120 2374 3034 1006 22309 1007 3410 3
792 12915 2484 1516 2184 2000 7796 2037 2353 3565 4605 2516 1012 1996 2208 20
01 2209 2006 2337 1021 1010 2355 1010 2012 11902 1005 1055 3346 1999 1996 262
4 3799 3016 2181 2012 4203 10254 1010 2662 1012 2004 2023 2001 1996 12951 356
5 4605 1010 1996 2223 13155 1996 1000 3585 5315 1000 2007 2536 2751 1011 1177
3 11107 1010 2004 2092 2004 8184 28324 2075 1996 4535 1997 10324 2169 3565 46
05 2208 2007 3142 16371 28990 2015 1006 2104 2029 1996 2208 2052 2031 2042 21
24 2004 1000 3565 4605 1048 1000 1007 1010 2061 2008 1996 8154 2071 14500 344
4 1996 5640 16371 28990 2015 2753 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.314069 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.315055 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.323262 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.325162 140319166060416 run_squad.py:432] unique_id: 1000000015
I0825 16:30:02.326185 140319166060416 run_squad.py:433] example_index: 15
I0825 16:30:02.327186 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.328224 140319166060416 run_squad.py:436] tokens: [CLS] super b
owl 50 decided the nfl champion for what season ? [SEP] super bowl 50 was an
 american football game to determine the champion of the national football le
ague ( nfl ) for the 2015 season . the american football conference ( afc ) c
hampion denver broncos defeated the national football conference ( nfc ) cham
pion carolina panthers 24 – 10 to earn their third super bowl title . the gam
e was played on february 7 , 2016 , at levi ' s stadium in the san francisco
 bay area at santa clara , california . as this was the 50th super bowl , the
league emphasized the " golden anniversary " with various gold - themed initi
atives , as well as temporarily suspend ##ing the tradition of naming each su
per bowl game with roman nu ##meral ##s ( under which the game would have bee
n known as " super bowl l " ) , so that the logo could prominently feature th
e arabic nu ##meral ##s 50 . [SEP]
I0825 16:30:02.329403 140319166060416 run_squad.py:438] token_to_orig_map: 1
3:0 14:1 15:2 16:3 17:4 18:5 19:6 20:7 21:8 22:9 23:10 24:11 25:12 26:13 27:1
4 28:15 29:16 30:17 31:17 32:17 33:18 34:19 35:20 36:21 37:21 38:22 39:23 40:
24 41:25 42:26 43:26 44:26 45:27 46:28 47:29 48:30 49:31 50:32 51:33 52:34 5
3:35 54:35 55:35 56:36 57:37 58:38 59:39 60:39 61:39 62:40 63:41 64:42 65:43
 66:44 67:45 68:46 69:46 70:47 71:48 72:49 73:50 74:51 75:52 76:53 77:53 78:5
4 79:54 80:55 81:56 82:56 83:56 84:57 85:58 86:59 87:60 88:61 89:62 90:63 91:
64 92:65 93:66 94:66 95:67 96:67 97:68 98:69 99:70 100:71 101:72 102:73 103:7
```

```
4 104:74 105:75 106:76 107:77 108:78 109:79 110:79 111:80 112:80 113:81 114:8
2 115:83 116:83 117:83 118:84 119:84 120:85 121:86 122:87 123:88 124:89 125:8
9 126:90 127:91 128:92 129:93 130:94 131:95 132:96 133:97 134:98 135:99 136:1
00 137:100 138:100 139:101 140:101 141:102 142:103 143:104 144:105 145:106 14
6:107 147:108 148:109 149:110 150:110 151:111 152:112 153:112 154:112 155:112
156:113 157:114 158:115 159:116 160:117 161:118 162:119 163:120 164:121 165:1
22 166:122 167:122 168:123 169:123
I0825 16:30:02.330446 140319166060416 run_squad.py:440] token_is_max_context:
13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:True 22:Tr
ue 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:True 31:True 3
2:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 40:True 41:Tru
e 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:True 51:
True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:True 60:True
 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:True 70:T
rue 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:True 79:True 8
0:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 88:True 89:Tru
e 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:True 99:
True 100:True 101:True 102:True 103:True 104:True 105:True 106:True 107:True
 108:True 109:True 110:True 111:True 112:True 113:True 114:True 115:True 116:
True 117:True 118:True 119:True 120:True 121:True 122:True 123:True 124:True
 125:True 126:True 127:True 128:True 129:True 130:True 131:True 132:True 133:
True 134:True 135:True 136:True 137:True 138:True 139:True 140:True 141:True
 142:True 143:True 144:True 145:True 146:True 147:True 148:True 149:True 150:
True 151:True 152:True 153:True 154:True 155:True 156:True 157:True 158:True
 159:True 160:True 161:True 162:True 163:True 164:True 165:True 166:True 167:
True 168:True 169:True
I0825 16:30:02.331570 140319166060416 run_squad.py:442] input_ids: 101 3565 4
605 2753 2787 1996 5088 3410 2005 2054 2161 1029 102 3565 4605 2753 2001 2019
2137 2374 2208 2000 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 20
05 1996 2325 2161 1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169 32
49 1996 2120 2374 3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000 77
96 2037 2353 3565 4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355
1010 2012 11902 1005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254
 1010 2662 1012 2004 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 1996
1000 3585 5315 1000 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184
 28324 2075 1996 4535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 20
15 1006 2104 2029 1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000
1007 1010 2061 2008 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2753
1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.332611 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.333716 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
I0825 16:30:02.342068 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.344041 140319166060416 run_squad.py:432] unique_id: 1000000016
I0825 16:30:02.345493 140319166060416 run_squad.py:433] example_index: 16
I0825 16:30:02.346425 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.347523 140319166060416 run_squad.py:436] tokens: [CLS] what ye
ar did the denver broncos secure a super bowl title for the third time ? [SE
P] super bowl 50 was an american football game to determine the champion of t
he national football league ( nfl ) for the 2015 season . the american footba
ll conference ( afc ) champion denver broncos defeated the national football
 conference ( nfc ) champion carolina panthers 24 – 10 to earn their third su
per bowl title . the game was played on february 7 , 2016 , at levi ' s stadi
um in the san francisco bay area at santa clara , california . as this was th
e 50th super bowl , the league emphasized the " golden anniversary " with var
ious gold - themed initiatives , as well as temporarily suspend ##ing the tra
dition of naming each super bowl game with roman nu ##meral ##s ( under which
the game would have been known as " super bowl l " ) , so that the logo could
prominently feature the arabic nu ##meral ##s 50 . [SEP]
I0825 16:30:02.348666 140319166060416 run_squad.py:438] token_to_orig_map: 1
8:0 19:1 20:2 21:3 22:4 23:5 24:6 25:7 26:8 27:9 28:10 29:11 30:12 31:13 32:1
4 33:15 34:16 35:17 36:17 37:17 38:18 39:19 40:20 41:21 42:21 43:22 44:23 45:
24 46:25 47:26 48:26 49:26 50:27 51:28 52:29 53:30 54:31 55:32 56:33 57:34 5
8:35 59:35 60:35 61:36 62:37 63:38 64:39 65:39 66:39 67:40 68:41 69:42 70:43
 71:44 72:45 73:46 74:46 75:47 76:48 77:49 78:50 79:51 80:52 81:53 82:53 83:5
4 84:54 85:55 86:56 87:56 88:56 89:57 90:58 91:59 92:60 93:61 94:62 95:63 96:
64 97:65 98:66 99:66 100:67 101:67 102:68 103:69 104:70 105:71 106:72 107:73
 108:74 109:74 110:75 111:76 112:77 113:78 114:79 115:79 116:80 117:80 118:81
119:82 120:83 121:83 122:83 123:84 124:84 125:85 126:86 127:87 128:88 129:89
 130:89 131:90 132:91 133:92 134:93 135:94 136:95 137:96 138:97 139:98 140:99
141:100 142:100 143:100 144:101 145:101 146:102 147:103 148:104 149:105 150:1
06 151:107 152:108 153:109 154:110 155:110 156:111 157:112 158:112 159:112 16
0:112 161:113 162:114 163:115 164:116 165:117 166:118 167:119 168:120 169:121
170:122 171:122 172:122 173:123 174:123
I0825 16:30:02.349871 140319166060416 run_squad.py:440] token_is_max_context:
18:True 19:True 20:True 21:True 22:True 23:True 24:True 25:True 26:True 27:Tr
ue 28:True 29:True 30:True 31:True 32:True 33:True 34:True 35:True 36:True 3
7:True 38:True 39:True 40:True 41:True 42:True 43:True 44:True 45:True 46:Tru
e 47:True 48:True 49:True 50:True 51:True 52:True 53:True 54:True 55:True 56:
True 57:True 58:True 59:True 60:True 61:True 62:True 63:True 64:True 65:True
 66:True 67:True 68:True 69:True 70:True 71:True 72:True 73:True 74:True 75:T
rue 76:True 77:True 78:True 79:True 80:True 81:True 82:True 83:True 84:True 8
5:True 86:True 87:True 88:True 89:True 90:True 91:True 92:True 93:True 94:Tru
e 95:True 96:True 97:True 98:True 99:True 100:True 101:True 102:True 103:True
104:True 105:True 106:True 107:True 108:True 109:True 110:True 111:True 112:T
rue 113:True 114:True 115:True 116:True 117:True 118:True 119:True 120:True 1
21:True 122:True 123:True 124:True 125:True 126:True 127:True 128:True 129:Tr
ue 130:True 131:True 132:True 133:True 134:True 135:True 136:True 137:True 13
8:True 139:True 140:True 141:True 142:True 143:True 144:True 145:True 146:Tru
e 147:True 148:True 149:True 150:True 151:True 152:True 153:True 154:True 15
5:True 156:True 157:True 158:True 159:True 160:True 161:True 162:True 163:Tru
e 164:True 165:True 166:True 167:True 168:True 169:True 170:True 171:True 17
2:True 173:True 174:True
I0825 16:30:02.350978 140319166060416 run_squad.py:442] input_ids: 101 2054 2
095 2106 1996 7573 14169 5851 1037 3565 4605 2516 2005 1996 2353 2051 1029 10
2 3565 4605 2753 2001 2019 2137 2374 2208 2000 5646 1996 3410 1997 1996 2120
 2374 2223 1006 5088 1007 2005 1996 2325 2161 1012 1996 2137 2374 3034 1006 1
0511 1007 3410 7573 14169 3249 1996 2120 2374 3034 1006 22309 1007 3410 3792
 12915 2484 1516 2184 2000 7796 2037 2353 3565 4605 2516 1012 1996 2208 2001
```

```
 2209 2006 2337 1021 1010 2355 1010 2012 11902 1005 1055 3346 1999 1996 2624
 3799 3016 2181 2012 4203 10254 1010 2662 1012 2004 2023 2001 1996 12951 3565
4605 1010 1996 2223 13155 1996 1000 3585 5315 1000 2007 2536 2751 1011 11773
 11107 1010 2004 2092 2004 8184 28324 2075 1996 4535 1997 10324 2169 3565 460
5 2208 2007 3142 16371 28990 2015 1006 2104 2029 1996 2208 2052 2031 2042 212
4 2004 1000 3565 4605 1048 1000 1007 1010 2061 2008 1996 8154 2071 14500 3444
1996 5640 16371 28990 2015 2753 1012 102 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.352141 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.353271 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.361612 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.367515 140319166060416 run_squad.py:432] unique_id: 1000000017
I0825 16:30:02.368646 140319166060416 run_squad.py:433] example_index: 17
I0825 16:30:02.369644 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.370826 140319166060416 run_squad.py:436] tokens: [CLS] what ci
ty did super bowl 50 take place in ? [SEP] super bowl 50 was an american foot
ball game to determine the champion of the national football league ( nfl ) f
or the 2015 season . the american football conference ( afc ) champion denver
broncos defeated the national football conference ( nfc ) champion carolina p
anthers 24 – 10 to earn their third super bowl title . the game was played on
february 7 , 2016 , at levi ' s stadium in the san francisco bay area at sant
a clara , california . as this was the 50th super bowl , the league emphasize
d the " golden anniversary " with various gold - themed initiatives , as well
as temporarily suspend ##ing the tradition of naming each super bowl game wit
h roman nu ##meral ##s ( under which the game would have been known as " supe
r bowl l " ) , so that the logo could prominently feature the arabic nu ##mer
al ##s 50 . [SEP]
I0825 16:30:02.372376 140319166060416 run_squad.py:438] token_to_orig_map: 1
2:0 13:1 14:2 15:3 16:4 17:5 18:6 19:7 20:8 21:9 22:10 23:11 24:12 25:13 26:1
4 27:15 28:16 29:17 30:17 31:17 32:18 33:19 34:20 35:21 36:21 37:22 38:23 39:
24 40:25 41:26 42:26 43:26 44:27 45:28 46:29 47:30 48:31 49:32 50:33 51:34 5
2:35 53:35 54:35 55:36 56:37 57:38 58:39 59:39 60:39 61:40 62:41 63:42 64:43
 65:44 66:45 67:46 68:46 69:47 70:48 71:49 72:50 73:51 74:52 75:53 76:53 77:5
4 78:54 79:55 80:56 81:56 82:56 83:57 84:58 85:59 86:60 87:61 88:62 89:63 90:
64 91:65 92:66 93:66 94:67 95:67 96:68 97:69 98:70 99:71 100:72 101:73 102:74
103:74 104:75 105:76 106:77 107:78 108:79 109:79 110:80 111:80 112:81 113:82
 114:83 115:83 116:83 117:84 118:84 119:85 120:86 121:87 122:88 123:89 124:89
125:90 126:91 127:92 128:93 129:94 130:95 131:96 132:97 133:98 134:99 135:100
136:100 137:100 138:101 139:101 140:102 141:103 142:104 143:105 144:106 145:1
07 146:108 147:109 148:110 149:110 150:111 151:112 152:112 153:112 154:112 15
5:113 156:114 157:115 158:116 159:117 160:118 161:119 162:120 163:121 164:122
165:122 166:122 167:123 168:123
```

```
I0825 16:30:02.373493 140319166060416 run_squad.py:440] token_is_max_context:
12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:Tr
ue 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:True 3
1:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 40:Tru
e 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:
True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:True
 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:T
rue 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:True 7
9:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 88:Tru
e 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:
True 99:True 100:True 101:True 102:True 103:True 104:True 105:True 106:True 1
07:True 108:True 109:True 110:True 111:True 112:True 113:True 114:True 115:Tr
ue 116:True 117:True 118:True 119:True 120:True 121:True 122:True 123:True 12
4:True 125:True 126:True 127:True 128:True 129:True 130:True 131:True 132:Tru
e 133:True 134:True 135:True 136:True 137:True 138:True 139:True 140:True 14
1:True 142:True 143:True 144:True 145:True 146:True 147:True 148:True 149:Tru
e 150:True 151:True 152:True 153:True 154:True 155:True 156:True 157:True 15
8:True 159:True 160:True 161:True 162:True 163:True 164:True 165:True 166:Tru
e 167:True 168:True
I0825 16:30:02.374607 140319166060416 run_squad.py:442] input_ids: 101 2054 2
103 2106 3565 4605 2753 2202 2173 1999 1029 102 3565 4605 2753 2001 2019 2137
2374 2208 2000 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 2005 19
96 2325 2161 1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169 3249 19
96 2120 2374 3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000 7796 20
37 2353 3565 4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355 1010
2012 11902 1005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254 1010
 2662 1012 2004 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 1996 1000
3585 5315 1000 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184 28324
2075 1996 4535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 2015 1006
2104 2029 1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000 1007 10
10 2061 2008 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2753 1012 1
02 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.376127 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.377135 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.385693 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.387382 140319166060416 run_squad.py:432] unique_id: 1000000018
I0825 16:30:02.388947 140319166060416 run_squad.py:433] example_index: 18
I0825 16:30:02.390437 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.392016 140319166060416 run_squad.py:436] tokens: [CLS] what st
adium did super bowl 50 take place in ? [SEP] super bowl 50 was an american f
ootball game to determine the champion of the national football league ( nfl
```

) for the 2015 season . the american football conference (afc) champion de
nver broncos defeated the national football conference (nfc) champion carol
ina panthers 24 – 10 to earn their third super bowl title . the game was play
ed on february 7 , 2016 , at levi ' s stadium in the san francisco bay area a
t santa clara , california . as this was the 50th super bowl , the league emp
hasized the " golden anniversary " with various gold - themed initiatives , a
s well as temporarily suspend ##ing the tradition of naming each super bowl g
ame with roman nu ##meral ##s (under which the game would have been known as
" super bowl l ") , so that the logo could prominently feature the arabic nu
##meral ##s 50 . [SEP]
I0825 16:30:02.393761 140319166060416 run_squad.py:438] token_to_orig_map: 1
2:0 13:1 14:2 15:3 16:4 17:5 18:6 19:7 20:8 21:9 22:10 23:11 24:12 25:13 26:1
4 27:15 28:16 29:17 30:17 31:17 32:18 33:19 34:20 35:21 36:21 37:22 38:23 39:
24 40:25 41:26 42:26 43:26 44:27 45:28 46:29 47:30 48:31 49:32 50:33 51:34 5
2:35 53:35 54:35 55:36 56:37 57:38 58:39 59:39 60:39 61:40 62:41 63:42 64:43
65:44 66:45 67:46 68:46 69:47 70:48 71:49 72:50 73:51 74:52 75:53 76:53 77:5
4 78:54 79:55 80:56 81:56 82:56 83:57 84:58 85:59 86:60 87:61 88:62 89:63 90:
64 91:65 92:66 93:66 94:67 95:67 96:68 97:69 98:70 99:71 100:72 101:73 102:74
103:74 104:75 105:76 106:77 107:78 108:79 109:79 110:80 111:80 112:81 113:82
114:83 115:83 116:83 117:84 118:84 119:85 120:86 121:87 122:88 123:89 124:89
125:90 126:91 127:92 128:93 129:94 130:95 131:96 132:97 133:98 134:99 135:100
136:100 137:100 138:101 139:101 140:102 141:103 142:104 143:105 144:106 145:1
07 146:108 147:109 148:110 149:110 150:111 151:112 152:112 153:112 154:112 15
5:113 156:114 157:115 158:116 159:117 160:118 161:119 162:120 163:121 164:122
165:122 166:122 167:123 168:123
I0825 16:30:02.394831 140319166060416 run_squad.py:440] token_is_max_context:
12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:Tr
ue 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:True 3
1:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 40:Tru
e 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:
True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:True
60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:T
rue 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:True 7
9:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 88:Tru
e 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:
True 99:True 100:True 101:True 102:True 103:True 104:True 105:True 106:True 1
07:True 108:True 109:True 110:True 111:True 112:True 113:True 114:True 115:Tr
ue 116:True 117:True 118:True 119:True 120:True 121:True 122:True 123:True 12
4:True 125:True 126:True 127:True 128:True 129:True 130:True 131:True 132:Tru
e 133:True 134:True 135:True 136:True 137:True 138:True 139:True 140:True 14
1:True 142:True 143:True 144:True 145:True 146:True 147:True 148:True 149:Tru
e 150:True 151:True 152:True 153:True 154:True 155:True 156:True 157:True 15
8:True 159:True 160:True 161:True 162:True 163:True 164:True 165:True 166:Tru
e 167:True 168:True
I0825 16:30:02.396295 140319166060416 run_squad.py:442] input_ids: 101 2054 3
346 2106 3565 4605 2753 2202 2173 1999 1029 102 3565 4605 2753 2001 2019 2137
2374 2208 2000 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 2005 19
96 2325 2161 1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169 3249 19
96 2120 2374 3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000 7796 20
37 2353 3565 4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355 1010
2012 11902 1005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254 1010
2662 1012 2004 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 1996 1000
3585 5315 1000 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184 28324
2075 1996 4535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 2015 1006
2104 2029 1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000 1007 10
10 2061 2008 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2753 1012 1
02 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.399472 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.400977 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.412678 140319166060416 run_squad.py:431] *** Example ***
I0825 16:30:02.415291 140319166060416 run_squad.py:432] unique_id: 1000000019
I0825 16:30:02.417295 140319166060416 run_squad.py:433] example_index: 19
I0825 16:30:02.418764 140319166060416 run_squad.py:434] doc_span_index: 0
I0825 16:30:02.420556 140319166060416 run_squad.py:436] tokens: [CLS] what wa
s the final score of super bowl 50 ? [SEP] super bowl 50 was an american foot
ball game to determine the champion of the national football league ( nfl ) f
or the 2015 season . the american football conference ( afc ) champion denver
broncos defeated the national football conference ( nfc ) champion carolina p
anthers 24 – 10 to earn their third super bowl title . the game was played on
february 7 , 2016 , at levi ' s stadium in the san francisco bay area at sant
a clara , california . as this was the 50th super bowl , the league emphasize
d the " golden anniversary " with various gold - themed initiatives , as well
as temporarily suspend ##ing the tradition of naming each super bowl game wit
h roman nu ##meral ##s ( under which the game would have been known as " supe
r bowl l " ) , so that the logo could prominently feature the arabic nu ##mer
al ##s 50 . [SEP]
I0825 16:30:02.436574 140319166060416 run_squad.py:438] token_to_orig_map: 1
2:0 13:1 14:2 15:3 16:4 17:5 18:6 19:7 20:8 21:9 22:10 23:11 24:12 25:13 26:1
4 27:15 28:16 29:17 30:17 31:17 32:18 33:19 34:20 35:21 36:21 37:22 38:23 39:
24 40:25 41:26 42:26 43:26 44:27 45:28 46:29 47:30 48:31 49:32 50:33 51:34 5
2:35 53:35 54:35 55:36 56:37 57:38 58:39 59:39 60:39 61:40 62:41 63:42 64:43
 65:44 66:45 67:46 68:46 69:47 70:48 71:49 72:50 73:51 74:52 75:53 76:53 77:5
4 78:54 79:55 80:56 81:56 82:56 83:57 84:58 85:59 86:60 87:61 88:62 89:63 90:
64 91:65 92:66 93:66 94:67 95:67 96:68 97:69 98:70 99:71 100:72 101:73 102:74
103:74 104:75 105:76 106:77 107:78 108:79 109:79 110:80 111:80 112:81 113:82
 114:83 115:83 116:83 117:84 118:84 119:85 120:86 121:87 122:88 123:89 124:89
125:90 126:91 127:92 128:93 129:94 130:95 131:96 132:97 133:98 134:99 135:100
136:100 137:100 138:101 139:101 140:102 141:103 142:104 143:105 144:106 145:1
07 146:108 147:109 148:110 149:110 150:111 151:112 152:112 153:112 154:112 15
5:113 156:114 157:115 158:116 159:117 160:118 161:119 162:120 163:121 164:122
165:122 166:122 167:123 168:123
I0825 16:30:02.440453 140319166060416 run_squad.py:440] token_is_max_context:
12:True 13:True 14:True 15:True 16:True 17:True 18:True 19:True 20:True 21:Tr
ue 22:True 23:True 24:True 25:True 26:True 27:True 28:True 29:True 30:True 3
1:True 32:True 33:True 34:True 35:True 36:True 37:True 38:True 39:True 40:Tru
e 41:True 42:True 43:True 44:True 45:True 46:True 47:True 48:True 49:True 50:
True 51:True 52:True 53:True 54:True 55:True 56:True 57:True 58:True 59:True
 60:True 61:True 62:True 63:True 64:True 65:True 66:True 67:True 68:True 69:T
```

```
rue 70:True 71:True 72:True 73:True 74:True 75:True 76:True 77:True 78:True 7
9:True 80:True 81:True 82:True 83:True 84:True 85:True 86:True 87:True 88:Tru
e 89:True 90:True 91:True 92:True 93:True 94:True 95:True 96:True 97:True 98:
True 99:True 100:True 101:True 102:True 103:True 104:True 105:True 106:True 1
07:True 108:True 109:True 110:True 111:True 112:True 113:True 114:True 115:Tr
ue 116:True 117:True 118:True 119:True 120:True 121:True 122:True 123:True 12
4:True 125:True 126:True 127:True 128:True 129:True 130:True 131:True 132:Tru
e 133:True 134:True 135:True 136:True 137:True 138:True 139:True 140:True 14
1:True 142:True 143:True 144:True 145:True 146:True 147:True 148:True 149:Tru
e 150:True 151:True 152:True 153:True 154:True 155:True 156:True 157:True 15
8:True 159:True 160:True 161:True 162:True 163:True 164:True 165:True 166:Tru
e 167:True 168:True
I0825 16:30:02.447463 140319166060416 run_squad.py:442] input_ids: 101 2054 2
001 1996 2345 3556 1997 3565 4605 2753 1029 102 3565 4605 2753 2001 2019 2137
2374 2208 2000 5646 1996 3410 1997 1996 2120 2374 2223 1006 5088 1007 2005 19
96 2325 2161 1012 1996 2137 2374 3034 1006 10511 1007 3410 7573 14169 3249 19
96 2120 2374 3034 1006 22309 1007 3410 3792 12915 2484 1516 2184 2000 7796 20
37 2353 3565 4605 2516 1012 1996 2208 2001 2209 2006 2337 1021 1010 2355 1010
2012 11902 1005 1055 3346 1999 1996 2624 3799 3016 2181 2012 4203 10254 1010
 2662 1012 2004 2023 2001 1996 12951 3565 4605 1010 1996 2223 13155 1996 1000
3585 5315 1000 2007 2536 2751 1011 11773 11107 1010 2004 2092 2004 8184 28324
2075 1996 4535 1997 10324 2169 3565 4605 2208 2007 3142 16371 28990 2015 1006
2104 2029 1996 2208 2052 2031 2042 2124 2004 1000 3565 4605 1048 1000 1007 10
10 2061 2008 1996 8154 2071 14500 3444 1996 5640 16371 28990 2015 2753 1012 1
02 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.449646 140319166060416 run_squad.py:444] input_mask: 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
I0825 16:30:02.451709 140319166060416 run_squad.py:446] segment_ids: 0 0 0 0
 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

In [32]:

```
print(len(eval_examples))
print(len(eval_features))
```

```
10570
12006
```

In [33]:
```
tf.logging.info("***** Running predictions *****")
tf.logging.info("  Num orig examples = %d", len(eval_examples))
tf.logging.info("  Num features = %d", len(eval_features))

predict_input_fn = run_squad.input_fn_builder(
    input_file=eval_writer.filename,
    seq_length=MAX_SEQ_LENGTH,
    is_training=False,
    drop_remainder=False)
```

```
I0825 16:30:44.044410 140319166060416 <ipython-input-33-84a684c96aaa>:1] ****
* Running predictions *****
I0825 16:30:44.046566 140319166060416 <ipython-input-33-84a684c96aaa>:2]   Nu
m orig examples = 10570
I0825 16:30:44.047948 140319166060416 <ipython-input-33-84a684c96aaa>:3]   Nu
m features = 12006
```

In [34]:
```
all_results = []
for result in estimator.predict(predict_input_fn, yield_single_examples=True):
    if len(all_results) % 1000 == 0:
        tf.logging.info("Processing example: %d" % (len(all_results)))
    unique_id = int(result["unique_ids"])
    start_logits = [float(x) for x in result["start_logits"].flat]
    end_logits = [float(x) for x in result["end_logits"].flat]
    all_results.append(
          run_squad.RawResult(
              unique_id=unique_id,
              start_logits=start_logits,
              end_logits=end_logits))
```

```
I0825 16:30:44.345084 140319166060416 estimator.py:1145] Calling model_fn.
I0825 16:30:44.527903 140319166060416 run_squad.py:598] *** Features ***
I0825 16:30:44.529029 140319166060416 run_squad.py:600]   name = input_ids, s
hape = (1, 256)
I0825 16:30:44.529978 140319166060416 run_squad.py:600]   name = input_mask,
shape = (1, 256)
I0825 16:30:44.530787 140319166060416 run_squad.py:600]   name = segment_ids,
shape = (1, 256)
I0825 16:30:44.532435 140319166060416 run_squad.py:600]   name = unique_ids,
shape = (1,)
I0825 16:30:50.598454 140319166060416 run_squad.py:634] **** Trainable Variab
les ****
I0825 16:30:50.599766 140319166060416 run_squad.py:640]   name = bert/embeddi
ngs/word_embeddings:0, shape = (30522, 768), *INIT_FROM_CKPT*
I0825 16:30:50.600904 140319166060416 run_squad.py:640]   name = bert/embeddi
ngs/token_type_embeddings:0, shape = (2, 768), *INIT_FROM_CKPT*
I0825 16:30:50.601887 140319166060416 run_squad.py:640]   name = bert/embeddi
ngs/position_embeddings:0, shape = (512, 768), *INIT_FROM_CKPT*
I0825 16:30:50.602790 140319166060416 run_squad.py:640]   name = bert/embeddi
ngs/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.603598 140319166060416 run_squad.py:640]   name = bert/embeddi
ngs/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.604414 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.605305 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.606186 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.607062 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.607980 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.608881 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.609769 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:30:50.610676 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.611494 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.612370 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:30:50.613236 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:30:50.614100 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:30:50.614923 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:30:50.615818 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.616838 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.617747 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_0/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.618530 140319166060416 run_squad.py:640]   name = bert/encode
```

```
r/layer_1/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.619381 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.620289 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.621181 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.622128 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.623013 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.623935 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:30:50.624819 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.625722 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.626591 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:30:50.627705 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:30:50.628642 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:30:50.629422 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:30:50.630153 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.630946 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.631853 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_1/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.632775 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.633647 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.634500 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.635408 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.636332 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.637417 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.638301 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:30:50.639240 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.640130 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.641057 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:30:50.641943 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
```

```
I0825 16:30:50.642829 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:30:50.643731 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:30:50.644629 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.645560 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.646429 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_2/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.647361 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.648212 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.649165 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.650054 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.650924 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.651713 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.652588 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:30:50.653523 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.654448 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.655334 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:30:50.656270 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:30:50.657432 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:30:50.658369 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:30:50.659256 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.660189 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.661080 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_3/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.662033 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.662946 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.663869 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.664752 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.665706 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.666592 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.667490 140319166060416 run_squad.py:640]   name = bert/encode
```

```
r/layer_4/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:30:50.668350 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.669265 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.670121 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:30:50.671006 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:30:50.671893 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:30:50.672839 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:30:50.673706 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.674570 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.675491 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_4/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.676364 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.677463 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.678336 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.679265 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.680108 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.681073 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.681940 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:30:50.682920 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.683790 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.684775 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:30:50.685671 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:30:50.686593 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:30:50.687552 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:30:50.688421 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.689334 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.690227 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_5/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.691087 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
```

```
I0825 16:30:50.692008 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.692905 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.693831 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.694707 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.695671 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.696557 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:30:50.697643 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.698544 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.699495 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:30:50.700525 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:30:50.701451 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:30:50.702358 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:30:50.703314 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.704174 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.705050 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_6/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.705923 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.706880 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.707742 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.708669 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.709523 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.710447 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.711318 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:30:50.712227 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.713129 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.714062 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:30:50.714929 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:30:50.715780 140319166060416 run_squad.py:640]   name = bert/encode
```

```
r/layer_7/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:30:50.716856 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:30:50.717812 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.718677 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.719565 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_7/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.720490 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.721554 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.722537 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.723438 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.724336 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.725275 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.726180 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
T*
I0825 16:30:50.727037 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.727995 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.728899 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:30:50.729835 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:30:50.730702 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:30:50.731640 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:30:50.732537 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.733461 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.734360 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_8/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.735243 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.736143 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.737204 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.738102 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.738993 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.739948 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.740827 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CKP
```

```
T*
I0825 16:30:50.741750 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.742665 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.743592 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:30:50.744467 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:30:50.745402 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:30:50.746296 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:30:50.747177 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.748140 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.749018 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_9/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.749937 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT
*
I0825 16:30:50.750889 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.751754 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.752706 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.753589 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT
*
I0825 16:30:50.754521 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.755385 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CK
PT*
I0825 16:30:50.756393 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.757445 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:30:50.758500 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKP
T*
I0825 16:30:50.759377 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:30:50.760364 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:30:50.761257 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:30:50.762157 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.763085 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.763979 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_10/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
```

```
I0825 16:30:50.764928 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/self/query/kernel:0, shape = (768, 768), *INIT_FROM_CKPT
*
I0825 16:30:50.765816 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/self/query/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.766744 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/self/key/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.767641 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/self/key/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.768595 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/self/value/kernel:0, shape = (768, 768), *INIT_FROM_CKPT
*
I0825 16:30:50.769948 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/self/value/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.770797 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/output/dense/kernel:0, shape = (768, 768), *INIT_FROM_CK
PT*
I0825 16:30:50.771721 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.772609 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT
*
I0825 16:30:50.773530 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/attention/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKP
T*
I0825 16:30:50.774426 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/intermediate/dense/kernel:0, shape = (768, 3072), *INIT_FROM_CKPT*
I0825 16:30:50.775312 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/intermediate/dense/bias:0, shape = (3072,), *INIT_FROM_CKPT*
I0825 16:30:50.776217 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/output/dense/kernel:0, shape = (3072, 768), *INIT_FROM_CKPT*
I0825 16:30:50.777308 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/output/dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.778179 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/output/LayerNorm/beta:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.779035 140319166060416 run_squad.py:640]   name = bert/encode
r/layer_11/output/LayerNorm/gamma:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.779967 140319166060416 run_squad.py:640]   name = bert/pooler/
dense/kernel:0, shape = (768, 768), *INIT_FROM_CKPT*
I0825 16:30:50.780838 140319166060416 run_squad.py:640]   name = bert/pooler/
dense/bias:0, shape = (768,), *INIT_FROM_CKPT*
I0825 16:30:50.781760 140319166060416 run_squad.py:640]   name = cls/squad/ou
tput_weights:0, shape = (2, 768)
I0825 16:30:50.782659 140319166060416 run_squad.py:640]   name = cls/squad/ou
tput_bias:0, shape = (2,)
I0825 16:30:52.258410 140319166060416 estimator.py:1147] Done calling model_f
n.
I0825 16:30:52.266065 140319166060416 tpu_estimator.py:499] TPU job name work
er
I0825 16:30:52.791291 140319166060416 monitored_session.py:240] Graph was fin
alized.
I0825 16:30:55.567826 140319166060416 saver.py:1280] Restoring parameters fro
m gs://bert-on-squad/bert-checkpoints/models/SQUAD/model.ckpt-10949
I0825 16:31:08.317030 140319166060416 session_manager.py:500] Running local_i
nit_op.
I0825 16:31:08.505281 140319166060416 session_manager.py:502] Done running lo
cal_init_op.
```

```
I0825 16:31:08.927345 140319166060416 tpu_estimator.py:557] Init TPU system
I0825 16:31:17.359092 140319166060416 tpu_estimator.py:566] Initialized TPU i
n 8 seconds
I0825 16:31:17.361118 140316956202752 tpu_estimator.py:514] Starting infeed t
hread controller.
I0825 16:31:17.364986 140316947810048 tpu_estimator.py:533] Starting outfeed
thread controller.
I0825 16:31:17.633933 140319166060416 util.py:98] Initialized dataset iterato
rs in 0 seconds
I0825 16:31:17.807278 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:17.808620 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:17.811655 140316947810048 tpu_estimator.py:275] Outfeed finished
for iteration (0, 0)
I0825 16:31:22.850517 140319166060416 <ipython-input-34-d2faabfe1ad8>:4] Proc
essing example: 0
I0825 16:31:22.856201 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:22.857524 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:22.867501 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:22.868387 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:22.877237 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:22.878621 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:22.885711 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:22.887304 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:22.897598 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:22.898649 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:22.908593 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:22.909696 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:22.919993 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:22.921022 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:22.930388 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:22.931482 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:22.941996 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:22.943035 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:22.952392 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:22.953396 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
```

```
I0825 16:31:22.966880 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:22.967762 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:22.981543 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:22.982534 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:22.991373 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:22.993314 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.002701 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.003828 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.013139 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.014052 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.023390 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.024163 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.033506 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.034501 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.043830 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.044776 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.053140 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.054168 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.063483 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.064382 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.073140 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.074206 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.082535 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.083710 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.092493 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.093545 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.102559 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.103560 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.112502 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
batch(es) of data to infeed.
I0825 16:31:23.113469 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.119522 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.120844 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.129790 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.130646 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.140979 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.142022 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.150957 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.152029 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.160329 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.161206 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.170829 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.172557 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.182054 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.184308 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.193940 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.195164 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.207154 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.208401 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.218683 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.219975 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.229477 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.230574 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.241523 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.242511 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.254287 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.255924 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.268560 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
```

```
I0825 16:31:23.269733 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.280504 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.281626 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.291323 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.292290 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.301443 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.303013 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.318038 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.319313 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.330620 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.332262 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.352675 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.353897 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.365788 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.367091 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.383506 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.384817 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.398237 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.399582 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.415637 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.416940 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.430625 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.433444 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.445124 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.446182 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.459614 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.460747 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.470262 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.471573 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
batch(es) of data from outfeed.
I0825 16:31:23.483115 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.484302 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.494940 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.496131 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.507089 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.508289 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.519641 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.521687 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.542443 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.543893 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.556695 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.557899 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.568517 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.569977 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.580143 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.581205 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.593699 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.595066 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.605872 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.607173 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.635392 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.636638 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.646478 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.647718 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.660118 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.661830 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.677718 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.679096 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
```

```
I0825 16:31:23.691380 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.692383 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.701528 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.702689 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.714185 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.715738 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.738476 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.739855 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.750251 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.751507 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.765393 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.766455 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.778387 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.779659 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.793726 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.794938 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.807860 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.809299 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.821350 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.822453 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.836960 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.838770 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.851643 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.852962 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.864127 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.865262 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.877072 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.878211 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.890840 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
batch(es) of data to infeed.
I0825 16:31:23.891999 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.904766 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.906429 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.920165 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.921379 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.934877 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.936141 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.948400 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.949560 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.961673 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.962934 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.972729 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.974019 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:23.985793 140319166060416 tpu_estimator.py:590] Enqueue next (1)
batch(es) of data to infeed.
I0825 16:31:23.986852 140319166060416 tpu_estimator.py:594] Dequeue next (1)
batch(es) of data from outfeed.
I0825 16:31:24.002538 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.003763 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.016820 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.017974 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.031036 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.032378 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.048208 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.049384 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.061371 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.062546 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.076329 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.077778 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.093026 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:24.094539 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.108293 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.109578 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.125662 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.127034 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.140732 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.142134 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.153022 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.154168 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.165995 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.167177 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.186233 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.187674 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.198289 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.199384 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.208789 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.209953 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.221176 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.222305 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.234135 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.235306 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.247141 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.248260 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.258983 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.260180 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.279238 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.280645 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.296982 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.298566 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:24.315463 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.316786 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.331643 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.333193 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.346222 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.347449 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.360536 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.361791 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.372992 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.374221 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.391416 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.392595 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.404146 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.405306 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.416253 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.417370 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.430072 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.431275 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.444418 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.445550 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.455990 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.457132 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.469646 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.470788 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.483906 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.485177 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.500566 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.502018 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:24.515014 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.516246 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.525604 140319166060416 <ipython-input-34-d2faabfe1ad8>:4] Proc
essing example: 1000
I0825 16:31:24.531156 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.532100 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.542448 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.543442 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.554590 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.555541 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.565534 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.566760 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.578481 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.579738 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.593856 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.595221 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.608664 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.610521 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.629160 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.630352 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.645974 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.647549 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.660710 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.661977 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.672559 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.673630 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.684761 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.688561 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.705416 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.706664 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:24.720080 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.721454 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.737000 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.738169 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.749725 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.750950 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.763015 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.764771 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.781426 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.782901 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.800841 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.801863 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.812551 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.813551 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.824575 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.825521 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.839679 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.840775 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.851381 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.852308 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.868553 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.869571 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.880431 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.881329 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.893120 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.894265 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.908599 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.909543 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:24.926599 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.927605 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.941444 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.942485 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.953682 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.955340 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.967416 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.968302 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.980063 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.980926 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:24.991171 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:24.992255 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.003443 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.004369 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.014329 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.015323 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.025798 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.027292 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.037214 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.038253 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.051653 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.053428 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.062920 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.063874 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.073077 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.073968 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.092973 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.094961 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.110984 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:25.112240 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.125584 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.126496 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.141047 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.142033 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.154731 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.156233 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.170037 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.171051 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.181055 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.181966 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.206210 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.207480 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.223183 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.224330 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.248726 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.249843 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.262343 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.263387 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.277818 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.278687 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.291238 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.292410 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.304226 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.305276 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.316139 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.316980 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.328845 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:25.329758 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.340685 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.341697 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.358318 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.359321 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.374114 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.375262 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.389157 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.390066 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.403343 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.404509 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.419204 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.420306 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.435328 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.436370 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.455851 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.456860 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.471542 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.472509 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.482121 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.483075 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.496357 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.497395 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.507766 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.508827 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.521163 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.522106 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.534018 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.534983 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:25.546815 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.548098 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.559587 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.560720 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.572630 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.573589 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.584912 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.585778 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.597655 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.598749 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.628525 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.629749 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.642037 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.642946 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.653113 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.654001 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.666308 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.667384 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.680449 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.681473 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.692895 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.693879 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.708689 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.710148 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.722843 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.723886 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.736701 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.737793 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:25.751130 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.752060 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.763360 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.764535 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.782720 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.783769 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.813728 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.815056 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.825942 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.827314 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.839743 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.840863 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.852941 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.854164 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.871832 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.872907 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.889274 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.890522 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.905153 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.906238 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.916924 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.917783 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.928026 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.929122 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.941583 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.942869 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.952303 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.953616 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.964311 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:25.965689 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.979213 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.980372 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:25.991266 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:25.992321 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.004868 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.005792 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.021981 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.026193 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.044665 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.045667 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.062411 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.063430 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.073573 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.074510 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.087129 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.087989 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.098677 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.099690 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.116048 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.117089 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.136737 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.137890 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.155255 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.156450 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.176841 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.177868 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.189888 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:26.190663 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.201847 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.202795 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.213654 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.214611 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.229954 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.230978 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.241274 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.242314 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.256777 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.257851 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.276622 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.277681 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.292042 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.293145 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.307512 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.308414 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.317080 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.318596 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.335304 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.336308 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.347449 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.348410 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.363375 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.364446 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.372328 140319166060416 <ipython-input-34-d2faabfe1ad8>:4] Proc
essing example: 2000
I0825 16:31:26.376045 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.378037 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.391226 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:26.392416 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.404904 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.405875 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.439311 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.440709 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.450251 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.451177 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.464882 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.465742 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.473770 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.474758 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.488085 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.488922 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.506330 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.507310 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.517065 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.518208 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.528775 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.529955 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.547661 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.549097 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.582491 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.583453 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.598559 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.599474 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.622991 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.624018 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.632845 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:26.633784 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.645878 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.646936 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.657208 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.658201 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.667066 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.668161 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.677869 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.679122 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.692229 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.693228 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.707645 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.708949 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.720368 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.721369 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.731590 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.732783 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.745852 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.746951 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.759026 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.760274 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.769629 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.770610 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.782513 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.783966 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.796041 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.797279 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.810423 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.811648 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:26.823232 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.824865 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.836219 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.837214 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.851704 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.852725 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.865234 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.866279 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.881925 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.882952 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.893168 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.894057 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.909721 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.911063 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.927445 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.928359 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.938716 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.939518 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.954993 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.956035 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.967155 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.968180 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.979333 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.980319 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:26.990975 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:26.991837 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.002212 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.003037 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:27.014212 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.015190 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.030935 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.032036 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.043837 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.047204 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.057985 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.058822 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.073218 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.074116 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.092987 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.094594 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.114274 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.115280 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.124480 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.125509 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.135293 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.136243 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.146636 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.147580 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.157560 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.158509 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.169156 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.170530 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.184413 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.185426 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.197268 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.198312 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.211641 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:27.212629 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.222629 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.223490 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.235019 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.236076 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.246484 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.248053 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.260853 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.261913 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.271721 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.272740 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.285702 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.286704 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.297000 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.297965 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.310188 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.311185 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.321357 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.322150 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.330638 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.331554 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.340391 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.341208 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.349987 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.350975 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.360364 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.361359 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.369887 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:27.371274 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.380918 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.381957 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.391357 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.392228 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.402969 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.403874 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.414002 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.415035 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.424374 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.425461 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.434735 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.435901 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.447669 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.448852 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.458104 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.459116 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.468928 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.470053 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.480579 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.481607 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.490692 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.491636 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.502842 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.506129 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.516788 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.517786 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.530861 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.532333 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:27.552515 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.553520 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.568684 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.569681 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.579764 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.580641 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.592255 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.593634 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.604194 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.605558 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.615236 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.616503 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.627472 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.628379 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.640758 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.641740 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.652478 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.653322 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.662935 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.663738 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.674703 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.675517 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.684232 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.685000 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.696325 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.697290 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.710926 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.712919 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:27.724797 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.725678 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.736343 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.737245 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.762517 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.763662 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.773528 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.774549 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.786072 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.786978 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.796268 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.797066 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.805606 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.806928 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.819982 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.820946 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.831958 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.832935 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.843931 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.844875 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.855347 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.856353 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.864941 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.865950 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.877352 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.878345 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.887530 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.888429 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.895576 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:27.896401 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.905726 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.906596 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.921466 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.922478 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.934396 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.935342 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.945218 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.946140 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.958509 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.959420 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.974593 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.975954 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.988644 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:27.990149 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:27.999339 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.000241 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.012106 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.012992 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.023745 140319166060416 <ipython-input-34-d2faabfe1ad8>:4] Proc
essing example: 3000
I0825 16:31:28.026754 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.027848 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.039668 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.040783 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.050019 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.051083 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.058502 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.059391 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:28.076892 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.078163 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.090662 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.091852 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.101729 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.102715 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.116545 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.117925 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.128964 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.130210 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.138614 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.139610 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.147278 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.148234 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.161006 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.162077 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.176196 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.177466 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.189349 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.190747 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.201260 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.202284 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.215480 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.216401 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.226502 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.227400 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.236797 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.237745 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.251269 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:28.252599 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.266922 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.267881 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.277667 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.278980 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.289621 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.290393 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.307932 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.309194 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.320901 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.322053 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.331241 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.332437 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.353289 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.354231 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.369019 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.370249 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.381243 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.382098 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.391970 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.392946 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.401532 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.402858 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.414407 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.415360 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.428867 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.429894 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.438225 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:28.439302 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.453078 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.454298 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.463124 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.464010 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.471723 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.472630 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.484748 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.485839 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.495830 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.496900 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.514080 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.515301 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.527312 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.528538 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.541346 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.542438 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.551554 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.552828 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.563357 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.564466 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.576873 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.577901 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.587028 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.588235 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.626003 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.627211 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.636872 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.637952 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:28.650262 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.651447 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.661740 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.663015 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.673922 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.675045 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.683910 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.685052 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.695405 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.696400 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.705255 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.706303 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.720840 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.721845 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.731375 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.732318 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.742934 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.744297 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.754261 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.755327 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.764430 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.765706 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.775388 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.776485 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.787752 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.789440 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.801915 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.802999 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:28.814164 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.815315 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.825299 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.826445 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.837938 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.838957 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.849905 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.850832 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.864023 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.865670 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.878144 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.879229 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.897386 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.899039 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.912678 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.913768 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.934845 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.935884 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.950245 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.951311 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.960952 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.961857 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.972757 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.973644 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.986824 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:28.987902 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:28.999110 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.000102 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.013760 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:29.014892 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.025348 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.026968 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.042106 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.043223 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.056281 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.057360 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.068024 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.070882 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.087838 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.090972 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.103441 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.104585 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.116051 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.117091 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.129270 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.130480 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.140573 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.141530 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.150912 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.152529 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.163952 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.164798 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.177136 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.178076 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.188375 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.189832 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.200913 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:29.202249 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.213841 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.215430 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.227312 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.228563 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.239403 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.240696 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.251765 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.252934 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.265289 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.266439 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.277358 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.278587 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.290531 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.291495 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.304969 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.306223 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.325097 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.326254 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.337160 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.338680 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.353701 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.354885 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.368431 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.369990 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.382998 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.384565 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.399120 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.400755 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:29.413359 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.415061 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.427430 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.429420 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.444213 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.446270 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.458288 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.459605 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.470970 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.472159 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.487081 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.488043 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.496353 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.497365 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.506501 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.507681 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.518145 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.519236 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.532434 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.533521 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.544214 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.545340 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.553997 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.555624 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.567244 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.568309 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.580343 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.581481 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:29.592451 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.593509 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.617233 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.618493 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.631316 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.632420 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.643577 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.644561 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.655254 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.656244 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.665584 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.666463 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.675899 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.676767 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.686047 140319166060416 <ipython-input-34-d2faabfe1ad8>:4] Proc
essing example: 4000
I0825 16:31:29.689917 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.690862 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.703518 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.704584 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.715914 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.716795 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.725175 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.726193 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.734075 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.737964 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.748654 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.749643 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.770213 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.771239 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:29.780783 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.782218 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.792263 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.793404 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.805019 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.806252 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.822738 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.823827 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.834163 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.835147 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.844359 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.846065 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.857255 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.858331 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.869464 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.870580 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.885659 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.886842 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.898365 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.899556 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.911019 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.912114 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.921629 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.922602 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.932906 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.933898 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.943564 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.944644 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:29.953713 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.954898 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.969672 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.970962 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.984089 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.985233 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:29.994528 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:29.995577 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.004546 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.005489 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.022279 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.023610 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.036717 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.037756 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.048095 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.049157 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.060630 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.061787 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.073010 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.074065 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.084728 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.085644 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.092291 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.093108 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.105144 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.106154 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.117967 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.119235 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.130678 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:30.131693 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.141622 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.142736 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.154437 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.156019 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.167535 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.168561 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.181915 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.182915 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.191723 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.192746 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.203198 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.204189 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.213435 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.214361 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.224467 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.225493 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.241276 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.242401 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.254655 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.257031 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.272737 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.274662 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.288508 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.290011 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.303113 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.304488 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.316993 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:30.318418 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.329916 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.331193 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.344000 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.345480 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.360420 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.361948 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.378242 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.379691 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.391938 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.393604 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.405757 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.406913 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.418039 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.419122 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.430197 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.431508 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.444747 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.446109 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.456440 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.457411 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.469937 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.471089 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.481936 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.483066 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.494042 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.495052 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.507883 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.508768 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:30.519030 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.520538 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.533223 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.534287 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.547688 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.549054 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.560588 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.561950 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.571774 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.572911 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.583827 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.584857 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.595368 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.596297 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.606236 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.607604 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.616492 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.617415 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.629582 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.630565 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.643219 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.644086 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.654548 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.655566 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.668403 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.669314 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.678815 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.679626 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:30.686635 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.687535 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.698067 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.698945 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.709414 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.710689 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.721793 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.722707 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.733558 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.735057 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.745289 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.746099 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.756287 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.757351 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.765941 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.766839 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.777566 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.778733 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.788347 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.789318 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.799629 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.800508 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.810678 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.811604 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.820726 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.821706 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.830874 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.831764 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.842563 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:30.843585 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.890534 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.891698 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.905604 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.906608 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.918259 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.919348 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.930557 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.931431 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.941294 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.942278 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.950544 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.951534 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.961062 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.962017 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.976108 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.977177 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:30.986228 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:30.987218 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.000882 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.002793 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.015462 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.016913 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.029644 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.032616 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.042614 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.044075 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.053797 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:31.055140 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.067102 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.068099 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.080412 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.081623 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.094288 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.095243 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.105510 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.106542 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.116101 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.117164 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.125271 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.126207 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.135565 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.136492 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.149297 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.150297 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.160393 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.161395 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.171710 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.172724 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.182256 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.183268 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.191915 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.192740 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.202249 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.203223 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.215519 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.217055 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:31.227365 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.228705 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.236095 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.236960 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.248008 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.249032 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.262447 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.263496 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.269587 140319166060416 <ipython-input-34-d2faabfe1ad8>:4] Proc
essing example: 5000
I0825 16:31:31.273999 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.274919 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.285697 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.286617 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.295663 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.296843 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.308730 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.309665 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.321966 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.324983 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.335173 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.336381 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.346930 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.347781 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.359592 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.361124 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.374774 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.376540 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.392382 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:31.394279 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.406886 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.408240 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.418843 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.420496 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.433676 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.435309 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.446595 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.447762 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.461101 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.463489 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.478290 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.479514 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.492164 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.493552 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.504367 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.505683 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.514368 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.515326 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.524412 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.525728 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.536645 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.540010 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.550844 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.551916 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.564006 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.565068 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.574865 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.577533 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:31.590150 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.591335 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.608085 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.609639 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.620776 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.622456 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.633109 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.634194 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.644920 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.645961 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.657292 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.658423 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.667696 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.668738 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.676564 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.677633 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.690349 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.691545 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.706204 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.708558 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.721234 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.722441 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.734197 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.735751 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.749319 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.751280 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.765261 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.767250 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:31.781039 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.782495 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.796088 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.800442 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.818612 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.820104 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.830860 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.835645 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.848490 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.850328 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.864788 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.866376 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.879933 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.881119 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.894678 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.896294 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.909989 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.911523 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.925090 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.926387 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.937009 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.938416 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.949285 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.951088 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.960435 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.961951 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.973148 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:31.974494 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:31.985924 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:31.987355 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.003715 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.005034 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.015860 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.017176 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.030667 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.032027 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.042934 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.044294 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.055732 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.057123 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.069442 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.070898 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.085656 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.088475 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.099142 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.100760 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.113264 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.114971 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.126235 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.127892 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.139158 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.140602 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.151395 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.153294 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.165751 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.167330 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.178462 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:32.180043 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.195850 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.197140 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.215645 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.217581 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.232612 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.234250 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.244747 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.246829 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.258420 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.259837 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.271087 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.273014 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.284858 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.286300 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.295306 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.296573 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.306070 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.307380 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.317387 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.318824 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.328716 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.330628 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.342693 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.343979 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.356381 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.357684 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.368395 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.370247 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:32.390483 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.391879 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.402370 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.404310 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.416143 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.417572 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.431031 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.432460 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.443651 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.445128 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.457855 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.459120 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.471772 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.473985 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.486975 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.489002 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.499572 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.501646 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.510605 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.511772 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.521578 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.522944 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.533525 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.534886 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.545048 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.546311 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.556557 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.558460 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:32.569082 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.570980 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.581546 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.582870 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.591897 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.593840 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.604330 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.606176 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.616572 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.618557 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.629955 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.632406 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.643611 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.645730 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.657145 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.658426 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.668937 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.670215 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.679643 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.680914 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.690655 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.692123 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.709397 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.710771 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.722620 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.723966 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.733917 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.735775 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.746976 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:32.748380 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.757439 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.758901 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.767928 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.769372 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.779188 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.780487 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.789546 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.790906 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.801391 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.802694 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.811856 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.813228 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.822468 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.823627 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.834392 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.835679 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.846345 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.848445 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.858494 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.859891 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.870548 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.872349 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.882149 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.883468 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.893886 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.895213 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.904024 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:32.905250 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.914646 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.915863 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.922286 140319166060416 <ipython-input-34-d2faabfe1ad8>:4] Proc
essing example: 6000
I0825 16:31:32.927223 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.928258 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.937098 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.938467 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.951950 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.953620 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.964384 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.965826 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.976202 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.977528 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.987222 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.988566 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:32.998049 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:32.999549 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.008618 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.009952 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.018701 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.020125 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.033156 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.034660 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.046426 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.048086 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.057862 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.059568 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.069398 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:33.070906 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.080772 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.082377 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.094035 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.096629 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.107537 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.109020 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.118501 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.119874 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.128770 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.130935 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.141355 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.142739 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.152651 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.154728 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.166204 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.167966 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.178518 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.180044 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.194602 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.196990 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.207096 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.208710 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.220102 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.222586 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.233083 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.236191 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.246219 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:33.247687 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.257127 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.258724 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.267536 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.268977 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.280163 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.282225 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.291183 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.292979 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.301508 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.303792 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.313516 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.315276 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.336694 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.340954 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.350476 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.352084 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.362459 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.363978 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.372789 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.374275 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.383670 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.385038 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.398958 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.400466 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.409856 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.411316 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.420571 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.422192 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:33.431166 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.432543 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.441363 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.442653 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.451722 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.453330 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.468614 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.472641 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.483449 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.485041 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.498106 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.499581 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.515222 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.518354 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.530538 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.532159 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.541991 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.543418 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.553746 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.555486 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.567224 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.568511 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.578477 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.580144 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.592428 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.593787 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.604710 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.606249 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:33.616965 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.618244 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.628829 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.630311 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.639828 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.641296 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.651737 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.653175 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.661882 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.663946 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.674841 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.676160 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.686441 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.687736 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.696601 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.697912 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.708135 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.709410 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.717968 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.719409 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.731598 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.733953 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.744634 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.746165 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.755518 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.756939 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.766325 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.768106 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.777060 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:33.778481 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.787516 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.789054 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.798703 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.800175 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.809358 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.810765 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.820530 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.821992 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.830929 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.833783 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.844113 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.845310 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.853521 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.854771 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.864448 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.865926 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.874916 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.876456 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.887486 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.890718 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.900460 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.901927 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.912098 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.913599 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.922946 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.924287 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.933827 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:33.935337 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.946541 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.949955 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.960761 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.962241 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.971589 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.973422 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.983380 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.984884 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:33.994571 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:33.996010 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.006072 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.007622 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.016834 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.018208 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.027449 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.028912 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.038216 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.039729 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.050603 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.051975 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.061676 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.063064 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.072642 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.074047 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.084035 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.085372 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.095037 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.096390 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:34.105587 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.106782 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.115546 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.116763 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.125682 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.127128 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.136228 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.137410 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.146871 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.148496 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.157557 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.158750 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.169771 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.171325 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.184018 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.186041 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.198013 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.205344 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.220019 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.221400 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.231536 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.232928 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.241791 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.242989 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.251964 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.253623 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.264214 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.265748 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:34.275229 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.276854 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.286331 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.288239 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.298862 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.300144 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.309967 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.311253 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.320684 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.322111 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.331406 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.332579 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.341258 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.342480 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.351546 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.352851 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.362057 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.363440 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.382505 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.384163 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.396451 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.397722 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.407026 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.408481 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.420048 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.421571 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.429151 140319166060416 <ipython-input-34-d2faabfe1ad8>:4] Proc
essing example: 7000
I0825 16:31:34.436271 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.437391 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:34.446624 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.447881 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.457382 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.458696 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.468244 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.469728 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.479951 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.481346 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.491455 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.492928 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.502722 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.504161 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.515260 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.516535 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.526485 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.527868 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.537039 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.538431 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.547969 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.549428 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.558698 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.560416 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.569720 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.571089 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.579905 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.581710 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.591017 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.592406 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:34.604361 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.606433 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.622723 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.624450 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.634538 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.636029 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.645975 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.647539 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.656887 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.658405 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.668574 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.671045 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.681611 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.683447 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.693506 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.695307 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.709613 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.711041 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.721302 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.722782 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.732495 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.733894 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.743875 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.745288 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.754208 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.755611 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.764996 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.766492 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.776059 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:34.777535 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.786869 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.788158 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.797930 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.799437 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.809327 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.810778 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.820596 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.822035 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.831610 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.833127 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.842633 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.844010 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.853419 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.854757 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.863901 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.865238 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.875018 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.876337 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.885308 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.886588 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.895523 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.897505 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.908459 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.909869 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.925479 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.926956 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.936257 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:34.937603 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.947124 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.948446 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.958069 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.959312 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.969396 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.970761 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.981063 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.982363 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:34.991897 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:34.993149 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.002507 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.003927 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.013207 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.014654 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.058602 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.060094 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.069722 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.071098 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.080698 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.082011 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.091686 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.093256 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.103093 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.104336 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.114087 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.115409 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.125102 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.126461 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:35.136027 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.137417 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.150951 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.152341 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.162272 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.163794 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.173718 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.175164 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.185167 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.186460 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.196047 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.197279 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.207582 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.209117 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.219118 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.220403 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.229994 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.231432 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.241448 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.242960 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.254547 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.256020 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.265300 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.266525 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.275547 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.277024 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.291448 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.292863 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:35.302076 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.303653 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.315570 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.317109 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.329048 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.330470 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.345653 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.347209 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.358713 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.360207 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.370616 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.372065 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.385066 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.386450 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.396080 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.398155 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.409429 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.412854 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.421365 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.424116 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.432236 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.434389 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.444940 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.446222 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.456108 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.458131 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.469120 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.470597 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.480320 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:35.481542 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.491031 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.492401 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.502549 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.504507 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.514897 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.516591 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.527647 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.528989 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.537995 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.539240 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.548682 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.550318 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.560023 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.563234 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.574104 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.575428 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.583472 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.585260 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.596368 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.598551 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.610833 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.612699 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.622670 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.624505 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.639247 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.641385 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.652443 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:35.653863 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.666056 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.667712 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.677505 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.678672 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.687513 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.688682 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.697900 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.699266 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.715660 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.717842 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.729490 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.731235 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.742011 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.743170 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.751925 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.753099 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.763476 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.765391 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.777304 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.778713 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.787926 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.789645 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.799824 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.801648 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.812848 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.814759 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.826506 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.828519 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:35.838605 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.839996 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.850985 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.852213 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.860780 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.862174 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.871905 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.873229 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.883254 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.884680 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.894558 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.895826 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.904903 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.906141 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.917747 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.921141 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.936568 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.938283 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.947650 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.949074 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.955531 140319166060416 <ipython-input-34-d2faabfe1ad8>:4] Proc
essing example: 8000
I0825 16:31:35.961041 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.962268 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.971936 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.973243 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.982749 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:35.984108 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:35.995575 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:35.996746 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.005912 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.007094 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.016286 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.017665 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.026757 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.028926 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.041687 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.043096 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.056089 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.057410 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.066432 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.067662 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.076992 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.078340 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.087911 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.089159 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.097958 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.099107 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.107773 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.109182 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.119496 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.120826 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.129947 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.131232 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.141033 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.142256 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.150961 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.152090 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:36.160897 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.162068 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.175760 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.177147 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.190429 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.191662 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.207093 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.208639 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.218624 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.220040 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.229255 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.231378 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.242398 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.243993 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.253239 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.254564 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.263253 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.264417 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.273284 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.274723 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.284201 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.285556 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.299348 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.301362 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.311147 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.312454 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.320588 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.322597 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:36.332033 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.333687 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.343046 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.344590 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.354030 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.355422 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.365018 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.366384 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.380120 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.381564 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.393210 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.394613 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.403648 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.405153 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.413735 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.415053 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.423583 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.424913 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.433693 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.435140 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.443759 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.446797 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.458662 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.459794 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.469296 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.470628 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.479538 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.481238 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.490361 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:36.491912 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.501352 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.503103 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.513782 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.515234 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.524863 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.526212 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.536072 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.537680 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.547684 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.549263 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.558612 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.560100 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.570630 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.571830 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.581088 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.582500 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.592078 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.593347 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.604721 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.606138 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.617178 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.618989 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.627708 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.628991 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.638159 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.639521 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.649242 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:36.650580 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.659666 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.661046 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.669685 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.671502 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.681738 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.683166 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.697208 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.698711 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.709240 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.710604 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.719580 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.720744 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.729657 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.730866 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.740827 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.742260 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.752036 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.753398 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.762360 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.763613 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.772084 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.775966 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.786072 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.787599 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.797987 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.799640 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.809972 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.811191 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:36.823354 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.825046 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.837108 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.838524 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.850178 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.852632 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.865204 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.866503 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.875966 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.877162 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.886281 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.887731 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.898599 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.900699 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.912603 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.913979 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.923840 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.925008 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.933969 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.935146 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.944522 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.945844 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.955076 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.956263 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.965653 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.967028 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.975767 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.977133 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:36.985677 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.987941 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:36.998017 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:36.999412 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.012263 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.013695 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.022793 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.024187 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.033035 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.034228 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.044738 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.046219 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.055520 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.056885 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.068463 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.069772 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.078908 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.080138 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.089716 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.091007 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.100796 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.103887 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.112996 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.114853 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.125204 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.126454 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.135131 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.136714 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.146663 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:37.148329 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.159677 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.161187 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.171192 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.172443 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.182142 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.183420 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.193010 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.194592 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.205787 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.207347 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.217194 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.218788 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.228724 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.230175 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.239977 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.241371 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.251616 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.253312 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.262721 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.264266 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.274909 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.276642 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.287037 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.288460 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.297764 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.299157 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.308364 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:37.309735 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.320686 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.322254 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.332518 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.334108 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.350429 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.352159 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.362654 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.364126 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.373980 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.375603 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.386180 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.387446 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.397640 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.399200 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.406452 140319166060416 <ipython-input-34-d2faabfe1ad8>:4] Proc
essing example: 9000
I0825 16:31:37.412918 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.414772 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.424776 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.426226 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.436330 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.437663 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.447708 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.449166 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.458719 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.460176 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.470334 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.471891 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.483037 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:37.484286 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.494526 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.498364 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.517263 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.518697 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.529553 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.530924 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.541089 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.542308 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.552115 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.553327 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.563388 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.565316 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.576599 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.578166 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.588369 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.590893 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.601736 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.603350 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.612437 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.614084 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.623694 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.625238 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.636613 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.639327 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.649272 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.650881 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.661701 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:37.665063 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.675102 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.676409 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.688028 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.689261 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.698771 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.700703 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.714889 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.716363 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.726066 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.727416 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.737101 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.738423 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.747623 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.748835 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.758215 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.759562 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.769441 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.770863 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.779755 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.781076 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.790763 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.795375 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.806173 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.807831 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.819115 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.820594 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.830420 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.831919 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:37.843389 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.844893 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.855222 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.856796 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.866693 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.868132 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.878035 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.879690 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.889484 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.891736 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.902371 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.904071 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.913657 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.915126 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.926606 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.929235 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.939527 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.941091 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.950377 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.952007 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.961955 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.963558 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.973510 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.975247 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.985103 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.986829 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:37.996205 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:37.997718 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:38.007703 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.010005 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.020857 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.023193 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.037578 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.039221 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.048645 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.050097 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.059868 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.061372 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.070765 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.072247 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.081409 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.083053 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.092242 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.093737 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.103180 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.105228 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.118758 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.120266 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.130496 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.134187 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.143862 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.145347 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.155843 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.157445 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.167012 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.168530 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.177595 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:38.179018 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.188517 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.190223 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.206828 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.208423 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.218589 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.220139 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.230245 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.231873 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.241551 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.243202 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.254334 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.255959 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.265587 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.267115 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.276486 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.277984 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.287590 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.289043 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.298868 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.300270 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.310728 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.312140 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.322285 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.323876 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.334894 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.336495 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.347497 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:38.349194 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.362865 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.364318 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.374513 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.376148 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.392123 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.393505 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.405040 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.406555 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.416846 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.418239 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.428994 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.430406 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.443062 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.444782 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.457007 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.458308 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.468489 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.469906 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.479559 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.480901 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.490480 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.491903 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.501689 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.503069 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.513587 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.515361 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.525431 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.526954 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:38.536455 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.538079 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.547532 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.549331 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.561709 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.563036 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.572549 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.573966 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.583540 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.585194 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.595507 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.597285 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.607774 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.609208 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.620667 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.622904 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.637051 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.638608 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.649219 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.650724 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.661520 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.663078 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.679689 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.681489 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.692522 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.693998 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.704715 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.706256 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:38.716614 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.718058 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.730784 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.732047 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.742220 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.743548 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.753778 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.755207 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.765322 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.766887 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.776967 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.778191 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.787558 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.789036 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.798480 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.800224 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.810055 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.811590 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.821855 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.823347 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.833593 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.835278 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.845140 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.846373 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.856470 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.857845 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.867589 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.869096 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.878985 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:38.880462 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.890025 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.891433 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.901105 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.902760 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.912384 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.914036 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.923959 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.925396 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.932261 140319166060416 <ipython-input-34-d2faabfe1ad8>:4] Proc
essing example: 10000
I0825 16:31:38.938203 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.939772 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.949044 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.950606 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.960183 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.961501 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.971084 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.972460 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.982184 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.983480 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:38.993129 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:38.994354 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.004374 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.005749 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.014907 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.016302 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.026065 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.027456 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:39.037678 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.039060 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.080279 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.082001 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.091786 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.093507 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.102733 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.110761 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.120518 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.121827 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.132128 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.133553 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.143517 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.144825 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.154237 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.155407 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.165414 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.166956 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.177578 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.179236 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.190059 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.191318 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.200871 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.202164 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.211440 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.212762 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.222947 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.224494 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.237132 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:39.239014 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.251286 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.252511 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.262691 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.267153 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.284453 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.289467 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.298967 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.300466 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.311278 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.313294 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.323868 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.325467 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.337966 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.339329 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.349437 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.350745 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.361207 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.362437 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.372399 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.373862 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.389984 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.391497 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.407827 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.410537 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.421362 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.422724 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.433814 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:39.435383 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.449460 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.451091 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.461615 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.462992 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.473613 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.477001 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.488013 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.489283 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.504166 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.505667 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.516420 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.517643 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.528869 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.530056 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.541000 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.542530 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.554557 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.556176 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.566512 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.568040 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.578142 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.579701 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.592756 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.594393 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.607922 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.609517 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.627085 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.628526 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:39.642556 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.646142 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.660572 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.662071 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.675742 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.677242 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.689475 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.691155 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.703840 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.705512 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.718286 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.719970 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.738070 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.739848 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.753740 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.757761 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.772848 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.774909 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.787843 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.789300 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.801379 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.802991 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.815469 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.817212 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.830760 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.832243 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.845382 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.847065 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:39.862522 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.864930 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.879007 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.880425 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.894275 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.895858 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.915826 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.917588 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.931603 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.933207 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.945253 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.946622 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.961794 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.963367 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.979335 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.980958 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:39.992470 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:39.993954 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.006629 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.008203 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.021116 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.022663 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.035564 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.037150 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.050086 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.051577 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.064500 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.066364 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.078694 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:40.080423 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.091859 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.093300 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.105262 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.106872 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.119083 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.120595 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.132329 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.134710 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.147622 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.150099 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.162415 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.164985 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.178184 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.179860 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.193022 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.194874 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.215198 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.216445 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.228110 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.229523 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.240405 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.242020 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.255430 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.256942 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.269854 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.271569 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.283652 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:40.285309 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.301922 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.303421 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.315624 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.317193 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.328225 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.329540 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.341327 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.342971 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.354007 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.355336 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.367357 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.368874 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.380158 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.381579 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.393756 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.395689 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.407204 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.408551 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.420135 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.421514 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.434142 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.435563 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.447313 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.448786 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.460368 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.461862 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.474153 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.475515 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:40.488135 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.489605 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.501051 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.502349 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.514505 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.516051 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.529465 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.531430 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.543653 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.545469 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.558284 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.559666 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.571436 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.572899 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.586077 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.587453 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.600385 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.602052 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.614121 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.615601 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.628211 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.633701 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.647315 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.649037 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.662175 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.664097 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.676177 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.677665 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:40.693585 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.695625 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.713434 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.715638 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.725600 140319166060416 <ipython-input-34-d2faabfe1ad8>:4] Proc
essing example: 11000
I0825 16:31:40.732389 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.733888 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.748486 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.750037 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.765932 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.767576 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.781615 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.783179 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.795176 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.796647 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.809043 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.810646 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.822571 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.824360 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.839020 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.840487 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.854749 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.856192 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.869558 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.871262 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.885329 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.887155 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.902566 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.904123 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:40.921010 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.922591 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.939165 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.940427 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.958207 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.959649 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.977432 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.979086 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:40.994030 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:40.995548 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.007605 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.009129 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.021268 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.023170 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.036498 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.038048 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.049956 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.051473 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.067032 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.068537 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.081196 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.082591 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.094792 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.096374 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.108998 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.110365 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.123060 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.124614 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:41.145143 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.146444 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.159200 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.161082 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.174324 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.175771 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.188525 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.190090 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.201692 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.203178 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.214544 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.216033 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.228252 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.229890 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.242560 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.244113 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.256240 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.257454 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.269651 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.271223 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.283170 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.284916 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.297181 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.301392 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.315304 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.316879 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.331011 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.332492 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.345365 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:41.347194 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.364751 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.366176 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.382184 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.383926 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.402271 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.406426 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.424042 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.425472 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.441305 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.442672 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.459145 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.460601 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.477700 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.479375 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.496521 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.497979 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.510488 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.512948 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.525113 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.526674 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.546033 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.548047 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.560303 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.561724 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.581870 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.583393 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.599475 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:41.603281 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.619289 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.620647 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.638165 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.639597 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.653366 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.654718 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.667827 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.669239 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.684029 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.685429 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.703524 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.705128 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.722569 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.724169 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.736900 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.738764 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.749932 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.751224 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.762205 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.763496 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.775227 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.776655 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.788862 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.790174 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.801884 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.803301 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.814851 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.816151 140319166060416 tpu_estimator.py:594] Dequeue next (1)
```

```
 batch(es) of data from outfeed.
I0825 16:31:41.828882 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.830391 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.842138 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.843599 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.854769 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.856203 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.867566 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.868931 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.880441 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.881873 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.892985 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.894344 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.907198 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.908678 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.920453 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.921839 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.936439 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.938172 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.951178 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.953411 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.966074 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.967246 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.984551 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.986034 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:41.997134 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:41.998707 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.012359 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.013987 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
```

```
I0825 16:31:42.026329 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.027997 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.043481 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.045370 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.057979 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.061871 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.074304 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.076287 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.092546 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.094146 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.105786 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.107312 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.127399 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.128783 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.141205 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.143901 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.159710 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.161280 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.173774 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.175109 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.189696 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.191106 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.208193 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.211700 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.223697 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.227217 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.240434 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.241778 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.254740 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:42.256180 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.267320 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.268639 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.280449 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.281742 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.294253 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.296072 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.307921 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.309686 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.320772 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.322185 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.334336 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.335712 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.347792 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.349630 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.364091 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.365504 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.378441 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.379859 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.391672 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.394107 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.408933 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.410622 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.424319 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.425943 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.438386 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.439983 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.453109 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
```

```
I0825 16:31:42.454474 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.467328 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.470007 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.481744 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.484026 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.498517 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.499979 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.514153 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.516640 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.529289 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.531065 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.545827 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.547309 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.560613 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.562086 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.576533 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.577987 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.590754 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.592754 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.613048 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.614554 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.630336 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.631692 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.644587 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.646045 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.661264 140319166060416 tpu_estimator.py:590] Enqueue next (1)
 batch(es) of data to infeed.
I0825 16:31:42.662437 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.672655 140319166060416 <ipython-input-34-d2faabfe1ad8>:4] Proc
essing example: 12000
I0825 16:31:42.677926 140319166060416 tpu_estimator.py:590] Enqueue next (1)
```

```
 batch(es) of data to infeed.
I0825 16:31:42.679479 140319166060416 tpu_estimator.py:594] Dequeue next (1)
 batch(es) of data from outfeed.
I0825 16:31:42.690238 140319166060416 tpu_estimator.py:598] Stop infeed threa
d controller
I0825 16:31:42.691655 140319166060416 tpu_estimator.py:430] Shutting down Inf
eedController thread.
I0825 16:31:42.693060 140316956202752 tpu_estimator.py:425] InfeedController
 received shutdown signal, stopping.
I0825 16:31:42.696302 140316956202752 tpu_estimator.py:530] Infeed thread fin
ished, shutting down.
I0825 16:31:42.697695 140319166060416 error_handling.py:96] infeed marked as
 finished
I0825 16:31:42.704353 140319166060416 tpu_estimator.py:602] Stop output threa
d controller
I0825 16:31:42.706919 140319166060416 tpu_estimator.py:430] Shutting down Out
feedController thread.
I0825 16:31:42.708286 140316947810048 tpu_estimator.py:425] OutfeedController
received shutdown signal, stopping.
I0825 16:31:42.713658 140316947810048 tpu_estimator.py:541] Outfeed thread fi
nished, shutting down.
I0825 16:31:42.715302 140319166060416 error_handling.py:96] outfeed marked as
finished
I0825 16:31:42.722773 140319166060416 tpu_estimator.py:606] Shutdown TPU syst
em.
I0825 16:31:43.104562 140319166060416 error_handling.py:96] prediction_loop m
arked as finished
I0825 16:31:43.106031 140319166060416 error_handling.py:96] prediction_loop m
arked as finished
```

In [0]:
```
output_prediction_file = os.path.join(OUTPUT_DIR, "predictions.json")
output_nbest_file = os.path.join(OUTPUT_DIR, "nbest_predictions.json")
output_null_log_odds_file = os.path.join(OUTPUT_DIR, "null_odds.json")
```

In [36]:
```
import collections
import json

write_predictions(eval_examples, eval_features, all_results,
                      20, 30,
                      True, output_prediction_file,
                      output_nbest_file, output_null_log_odds_file)
```

```
I0825 16:31:43.142138 140319166060416 <ipython-input-13-71425442c860>:5] Writ
ing predictions to: gs://bert-on-squad/bert-checkpoints/models/SQUAD/predicti
ons.json
I0825 16:31:43.145329 140319166060416 <ipython-input-13-71425442c860>:6] Writ
ing nbest to: gs://bert-on-squad/bert-checkpoints/models/SQUAD/nbest_predicti
ons.json
I0825 16:31:55.872426 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'German:[ˈmaɐ̯tiːnˈlʊtɐ](listen);10Novem
ber1483–18February1546)' vs 'german:[ˈmaɐtiːnˈlʊtɐ](listen);10november1483–18
february1546)'
I0825 16:31:55.874499 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'German:[ˈmaɐ̯tiːnˈlʊtɐ](listen);10Novem
ber1483–18February1546)' vs 'german:[ˈmaɐtiːnˈlʊtɐ](listen);10november1483–18
february1546)'
I0825 16:32:14.082412 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'togrog ( [UNK] )' in 'tögrög (₮).'
I0825 16:32:14.085624 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: '500 , 1 , 000 , 5 , 000 , 10 , 000 , and 20 , 000 mongolia
n togrog ( [UNK] )' in '500, 1,000, 5,000, 10,000, and 20,000 Mongolian tögrö
g (₮).'
I0825 16:32:14.087920 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'togrog ( [UNK] ) .' in 'tögrög (₮).'
I0825 16:32:14.089365 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: '20 , 000 mongolian togrog ( [UNK] )' in '20,000 Mongolian
tögrög (₮).'
I0825 16:32:14.090666 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'mongolian togrog ( [UNK] )' in 'Mongolian tögrög (₮).'
I0825 16:32:14.231104 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'chinese as simplified chinese : [UNK] 木 真' in 'Chinese as
simplified Chinese: 铁木真;'
I0825 16:32:14.232993 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'simplified chinese : [UNK] 木 真' in 'simplified Chinese:
铁木真;'
I0825 16:32:14.234751 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'chinese as simplified chinese : [UNK] 木 真 ; traditional c
hinese : [UNK] 木 [UNK] ; pinyin : tiemuzhen' in 'Chinese as simplified Chines
e: 铁木真; traditional Chinese: 鐵木眞; pinyin: Tiěmùzhēn.'
I0825 16:32:14.236253 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'simplified chinese : [UNK] 木 真 ; traditional chinese : [U
NK] 木 [UNK] ; pinyin : tiemuzhen' in 'simplified Chinese: 铁木真; traditional
Chinese: 鐵木眞; pinyin: Tiěmùzhēn.'
I0825 16:32:14.237544 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'as simplified chinese : [UNK] 木 真' in 'as simplified Chin
ese: 铁木真;'
I0825 16:32:14.238912 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'in chinese as simplified chinese : [UNK] 木 真' in 'in Chin
ese as simplified Chinese: 铁木真;'
I0825 16:32:14.240528 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'chinese as simplified chinese : [UNK] 木 真 ; traditional c
hinese : [UNK] 木 [UNK] ; pinyin : tiemuzhen .' in 'Chinese as simplified Chin
ese: 铁木真; traditional Chinese: 鐵木眞; pinyin: Tiěmùzhēn.'
I0825 16:32:14.241975 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'as simplified chinese : [UNK] 木 真 ; traditional chinese :
[UNK] 木 [UNK] ; pinyin : tiemuzhen' in 'as simplified Chinese: 铁木真; tradit
ional Chinese: 鐵木眞; pinyin: Tiěmùzhēn.'
I0825 16:32:14.243327 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'simplified chinese : [UNK] 木 真 ; traditional chinese : [U
```

```
NK] 木 [UNK] ; pinyin : tiemuzhen .' in 'simplified Chinese: 铁木真; tradition
al Chinese: 鐵木眞; pinyin: Tiěmùzhēn.'
I0825 16:32:14.245019 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'in chinese as simplified chinese : [UNK] 木 真 ; traditiona
l chinese : [UNK] 木 [UNK] ; pinyin : tiemuzhen' in 'in Chinese as simplified
Chinese: 铁木真; traditional Chinese: 鐵木眞; pinyin: Tiěmùzhēn.'
I0825 16:32:14.246347 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'chinese as simplified chinese : [UNK] 木 真 ; traditional c
hinese : [UNK] 木 [UNK]' in 'Chinese as simplified Chinese: 铁木真; traditiona
l Chinese: 鐵木眞;'
I0825 16:32:14.247717 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'chinese as simplified chinese : [UNK] 木 真 ;' in 'Chinese
as simplified Chinese: 铁木真;'
I0825 16:32:14.249100 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'simplified chinese : [UNK] 木 真 ; traditional chinese : [U
NK] 木 [UNK]' in 'simplified Chinese: 铁木真; traditional Chinese: 鐵木眞;'
I0825 16:32:14.250268 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'simplified chinese : [UNK] 木 真 ;' in 'simplified Chinese:
铁木真;'
I0825 16:32:14.251979 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'chinese as simplified chinese : [UNK] 木 真 ; traditional c
hinese : [UNK] 木 [UNK] ;' in 'Chinese as simplified Chinese: 铁木真; traditio
nal Chinese: 鐵木眞;'
I0825 16:32:14.253348 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'as simplified chinese : [UNK] 木 真 ; traditional chinese :
[UNK] 木 [UNK] ; pinyin : tiemuzhen .' in 'as simplified Chinese: 铁木真; trad
itional Chinese: 鐵木眞; pinyin: Tiěmùzhēn.'
I0825 16:32:14.254525 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'written in chinese as simplified chinese : [UNK] 木 真' in
'written in Chinese as simplified Chinese: 铁木真;'
I0825 16:32:14.255854 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'simplified chinese : [UNK] 木 真 ; traditional chinese : [U
NK] 木 [UNK] ;' in 'simplified Chinese: 铁木真; traditional Chinese: 鐵木眞;'
I0825 16:32:14.257182 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'in chinese as simplified chinese : [UNK] 木 真 ; traditiona
l chinese : [UNK] 木 [UNK] ; pinyin : tiemuzhen .' in 'in Chinese as simplifie
d Chinese: 铁木真; traditional Chinese: 鐵木眞; pinyin: Tiěmùzhēn.'
I0825 16:32:14.258487 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'as simplified chinese : [UNK] 木 真 ; traditional chinese :
[UNK] 木 [UNK]' in 'as simplified Chinese: 铁木真; traditional Chinese: 鐵木
眞;'
I0825 16:32:14.264850 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'chinghiz , chinghis , and chingiz , chinese : 成 吉 [UNK]
[UNK] ; pinyin : chengjisi han' in 'Chinghiz, Chinghis, and Chingiz, Chinese:
成吉思汗; pinyin: Chéngjísī Hán,'
I0825 16:32:14.270105 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'chinese : 成 吉 [UNK] [UNK] ; pinyin : chengjisi han' in 'C
hinese: 成吉思汗; pinyin: Chéngjísī Hán,'
I0825 16:32:14.920649 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'mortar and pestle and the [UNK] ( recipere ) character' in
'mortar and pestle and the ℞ (recipere) character,'
I0825 16:32:14.922381 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'mortar and pestle and the [UNK] ( recipere )' in 'mortar a
nd pestle and the ℞ (recipere)'
I0825 16:32:14.924237 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the mortar and pestle and the [UNK] ( recipere ) characte
r' in 'the mortar and pestle and the ℞ (recipere) character,'
```

```
I0825 16:32:14.925786 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the mortar and pestle and the [UNK] ( recipere )' in 'the
mortar and pestle and the ℞ (recipere)'
I0825 16:32:14.927121 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'mortar and pestle and the [UNK] ( recipere ) character ,'
in 'mortar and pestle and the ℞ (recipere) character,'
I0825 16:32:14.928549 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: '[UNK] ( recipere ) character' in '℞ (recipere) character,'
I0825 16:32:14.929779 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'mortar and pestle and the [UNK] ( recipere' in 'mortar and
pestle and the ℞ (recipere)'
I0825 16:32:14.930949 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'mortar and pestle and the [UNK]' in 'mortar and pestle and
the ℞'
I0825 16:32:14.932310 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'mortar and pestle and the [UNK] ( recipere ) character , w
hich is often written as " rx " in typed text .' in 'mortar and pestle and th
e ℞ (recipere) character, which is often written as "Rx" in typed text.'
I0825 16:32:14.933393 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the [UNK] ( recipere ) character' in 'the ℞ (recipere) cha
racter,'
I0825 16:32:14.935028 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the two symbols most commonly associated with pharmacy in
english - speaking countries are the mortar and pestle and the [UNK] ( recipe
re ) character' in 'The two symbols most commonly associated with pharmacy in
English-speaking countries are the mortar and pestle and the ℞ (recipere) cha
racter,'
I0825 16:32:14.936420 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the mortar and pestle and the [UNK] ( recipere ) character
,' in 'the mortar and pestle and the ℞ (recipere) character,'
I0825 16:32:14.937687 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'are the mortar and pestle and the [UNK] ( recipere ) chara
cter' in 'are the mortar and pestle and the ℞ (recipere) character,'
I0825 16:32:14.939180 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'two symbols most commonly associated with pharmacy in engl
ish - speaking countries are the mortar and pestle and the [UNK] ( recipere )
character' in 'two symbols most commonly associated with pharmacy in English-
speaking countries are the mortar and pestle and the ℞ (recipere) character,'
I0825 16:32:14.940305 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: '[UNK] ( recipere )' in '℞ (recipere)'
I0825 16:32:14.941834 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'pestle and the [UNK] ( recipere ) character' in 'pestle an
d the ℞ (recipere) character,'
I0825 16:32:14.947167 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: '[UNK] ( recipere ) character , which is often written as "
rx " in typed text . the show globe' in '℞ (recipere) character, which is oft
en written as "Rx" in typed text. The show globe'
I0825 16:32:14.949416 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'mortar and pestle and the [UNK] ( recipere ) character , w
hich is often written as " rx " in typed text . the show globe' in 'mortar an
d pestle and the ℞ (recipere) character, which is often written as "Rx" in ty
ped text. The show globe'
I0825 16:32:14.951577 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the mortar and pestle and the [UNK] ( recipere ) character
, which is often written as " rx " in typed text . the show globe' in 'the mo
rtar and pestle and the ℞ (recipere) character, which is often written as "R
x" in typed text. The show globe'
```

```
I0825 16:32:14.954553 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the [UNK] ( recipere ) character , which is often written
as " rx " in typed text . the show globe' in 'the ℞ (recipere) character, whi
ch is often written as "Rx" in typed text. The show globe'
I0825 16:32:17.986627 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Englishuniversitymodel—many' vs 'engli
shuniversitymodel—many'
I0825 16:32:17.988616 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'theEnglishuniversitymodel—many' vs 'th
eenglishuniversitymodel—many'
I0825 16:32:17.990175 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Englishuniversitymodel—many' vs 'engli
shuniversitymodel—many'
I0825 16:32:17.991763 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Englishuniversitymodel—manyleadersinth
ecolonyhadattendedtheUniversityofCambridge—but' vs 'englishuniversitymodel—ma
nyleadersinthecolonyhadattendedtheuniversityofcambridge—but'
I0825 16:32:17.993458 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Englishuniversitymodel—manyleadersinth
ecolonyhadattendedtheUniversityofCambridge—butconformedPuritanism.' vs 'engli
shuniversitymodel—manyleadersinthecolonyhadattendedtheuniversityofcambridge—b
utconformedpuritanism.'
I0825 16:32:17.994999 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'universitymodel—many' vs 'universitymo
del—many'
I0825 16:32:17.996270 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'ontheEnglishuniversitymodel—many' vs
'ontheenglishuniversitymodel—many'
I0825 16:32:17.998386 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Englishuniversitymodel—manyleadersinth
ecolonyhadattendedtheUniversityofCambridge—butconformedPuritanism.' vs 'engli
shuniversitymodel—manyleadersinthecolonyhadattendedtheuniversityofcambridge—b
utconformedpuritanism.'
I0825 16:32:17.999846 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'ItofferedaclassiccurriculumontheEnglis
huniversitymodel—many' vs 'itofferedaclassiccurriculumontheenglishuniversitym
odel—many'
I0825 16:32:18.001336 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Englishuniversitymodel—manyleadersinth
ecolonyhadattendedtheUniversityofCambridge—but' vs 'englishuniversitymodel—ma
nyleadersinthecolonyhadattendedtheuniversityofcambridge—but'
I0825 16:32:18.002970 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'classiccurriculumontheEnglishuniversit
ymodel—many' vs 'classiccurriculumontheenglishuniversitymodel—many'
I0825 16:32:18.004168 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'theEnglishuniversitymodel—many' vs 'th
eenglishuniversitymodel—many'
I0825 16:32:18.005209 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'model—many' vs 'model—many'
I0825 16:32:18.006769 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'aclassiccurriculumontheEnglishuniversi
tymodel—many' vs 'aclassiccurriculumontheenglishuniversitymodel—many'
I0825 16:32:18.008350 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'theEnglishuniversitymodel—manyleadersi
nthecolonyhadattendedtheUniversityofCambridge—but' vs 'theenglishuniversitymo
del—manyleadersinthecolonyhadattendedtheuniversityofcambridge—but'
I0825 16:32:22.407103 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
```

```
ble to find text: 'classic of changes ( i ching ) section regarding qian ( [U
NK] )' in 'Classic of Changes (I Ching) section regarding Qián (乾).'
I0825 16:32:22.410035 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'commentaries on the classic of changes ( i ching ) section
regarding qian ( [UNK] )' in 'Commentaries on the Classic of Changes (I Chin
g) section regarding Qián (乾).'
I0825 16:32:22.413380 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the commentaries on the classic of changes ( i ching ) sec
tion regarding qian ( [UNK] )' in 'the Commentaries on the Classic of Changes
(I Ching) section regarding Qián (乾).'
I0825 16:32:22.457559 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'xiao zhala ( [UNK] [UNK] [UNK] )' in 'Xiao Zhala (蕭札剌)'
I0825 16:32:22.459451 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shi tianze , liu heima ( 劉 [UNK] 馬 , liu ni ) , and the k
hitan xiao zhala' in 'Shi Tianze, Liu Heima (劉黑馬, Liu Ni), and the Khitan X
iao Zhala'
I0825 16:32:22.461363 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'liu heima ( 劉 [UNK] 馬 , liu ni ) , and the khitan xiao zh
ala' in 'Liu Heima (劉黑馬, Liu Ni), and the Khitan Xiao Zhala'
I0825 16:32:22.463216 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shi tianze , liu heima ( 劉 [UNK] 馬 , liu ni ) , and the k
hitan xiao zhala ( [UNK] [UNK] [UNK] )' in 'Shi Tianze, Liu Heima (劉黑馬, Liu
Ni), and the Khitan Xiao Zhala (蕭札剌)'
I0825 16:32:22.464699 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'khitan xiao zhala ( [UNK] [UNK] [UNK] )' in 'Khitan Xiao Z
hala (蕭札剌)'
I0825 16:32:22.466326 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'liu heima ( 劉 [UNK] 馬 , liu ni ) , and the khitan xiao zh
ala ( [UNK] [UNK] [UNK] )' in 'Liu Heima (劉黑馬, Liu Ni), and the Khitan Xiao
Zhala (蕭札剌)'
I0825 16:32:22.469126 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shi tianze , liu heima ( 劉 [UNK] 馬 , liu ni ) , and the k
hitan' in 'Shi Tianze, Liu Heima (劉黑馬, Liu Ni), and the Khitan'
I0825 16:32:22.470780 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'han chinese leaders , shi tianze , liu heima ( 劉 [UNK] 馬
, liu ni ) , and the khitan xiao zhala' in 'Han Chinese leaders, Shi Tianze,
Liu Heima (劉黑馬, Liu Ni), and the Khitan Xiao Zhala'
I0825 16:32:22.472097 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'liu heima ( 劉 [UNK] 馬 , liu ni ) , and the khitan' in 'Li
u Heima (劉黑馬, Liu Ni), and the Khitan'
I0825 16:32:22.475630 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shi tianze , liu heima ( 劉 [UNK] 馬 , liu ni ) , and the k
hitan xiao zhala' in 'Shi Tianze, Liu Heima (劉黑馬, Liu Ni), and the Khitan X
iao Zhala'
I0825 16:32:22.477348 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shi tianze , liu heima ( 劉 [UNK] 馬 , liu ni ) , and the k
hitan xiao zhala ( [UNK] [UNK] [UNK] )' in 'Shi Tianze, Liu Heima (劉黑馬, Liu
Ni), and the Khitan Xiao Zhala (蕭札剌)'
I0825 16:32:22.478624 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shi tianze , liu heima ( 劉 [UNK] 馬 , liu ni ) , and the k
hitan' in 'Shi Tianze, Liu Heima (劉黑馬, Liu Ni), and the Khitan'
I0825 16:32:22.479959 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'liu heima ( 劉 [UNK] 馬 , liu ni ) , and the khitan xiao zh
ala' in 'Liu Heima (劉黑馬, Liu Ni), and the Khitan Xiao Zhala'
I0825 16:32:22.481754 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shi tianze , liu heima ( 劉 [UNK] 馬 , liu ni )' in 'Shi Ti
anze, Liu Heima (劉黑馬, Liu Ni),'
```

```
I0825 16:32:22.483127 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'liu heima ( 劉 [UNK] 馬 , liu ni ) , and the khitan xiao zh
ala ( [UNK] [UNK] [UNK] )' in 'Liu Heima (劉黑馬, Liu Ni), and the Khitan Xiao
Zhala (蕭札剌)'
I0825 16:32:22.485022 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'xiao zhala ( [UNK] [UNK] [UNK] )' in 'Xiao Zhala (蕭札剌)'
I0825 16:32:22.486311 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shi tianze , liu heima ( 劉 [UNK] 馬 , liu ni ) , and the k
hitan xiao zhala (' in 'Shi Tianze, Liu Heima (劉黑馬, Liu Ni), and the Khitan
Xiao Zhala (蕭札剌)'
I0825 16:32:22.487593 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shi tianze , liu heima ( 劉 [UNK] 馬 , liu ni ) , and the k
hitan xiao' in 'Shi Tianze, Liu Heima (劉黑馬, Liu Ni), and the Khitan Xiao'
I0825 16:32:22.488856 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'liu heima ( 劉 [UNK] 馬 , liu ni ) , and the khitan' in 'Li
u Heima (劉黑馬, Liu Ni), and the Khitan'
I0825 16:32:22.489958 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'khitan xiao zhala ( [UNK] [UNK] [UNK] )' in 'Khitan Xiao Z
hala (蕭札剌)'
I0825 16:32:22.525350 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shi bingzhi ( 史 [UNK] [UNK] , shih ping - chih )' in 'Shi
Bingzhi (史秉直, Shih Ping-chih).'
I0825 16:32:22.527475 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shi bingzhi ( 史 [UNK] [UNK] , shih ping - chih ) .' in 'Sh
i Bingzhi (史秉直, Shih Ping-chih).'
I0825 16:32:22.529306 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shi bingzhi ( 史 [UNK] [UNK] , shih ping - chih ) . shi bin
gzhi' in 'Shi Bingzhi (史秉直, Shih Ping-chih). Shi Bingzhi'
I0825 16:32:22.531496 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shi bingzhi ( 史 [UNK]' in 'Shi Bingzhi (史秉直,'
I0825 16:32:22.533086 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shi bingzhi ( 史 [UNK] [UNK] , shih ping - chih ) . shi bin
gzhi was married to a jurchen woman' in 'Shi Bingzhi (史秉直, Shih Ping-chih).
Shi Bingzhi was married to a Jurchen woman'
I0825 16:32:22.535171 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'his father was shi bingzhi ( 史 [UNK] [UNK] , shih ping - c
hih )' in 'His father was Shi Bingzhi (史秉直, Shih Ping-chih).'
I0825 16:32:22.536496 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'bingzhi ( 史 [UNK] [UNK] , shih ping - chih )' in 'Bingzhi
(史秉直, Shih Ping-chih).'
I0825 16:32:22.537835 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'father was shi bingzhi ( 史 [UNK] [UNK] , shih ping - chih
)' in 'father was Shi Bingzhi (史秉直, Shih Ping-chih).'
I0825 16:32:22.539124 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'his father was shi bingzhi ( 史 [UNK] [UNK] , shih ping - c
hih ) .' in 'His father was Shi Bingzhi (史秉直, Shih Ping-chih).'
I0825 16:32:22.540541 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'was shi bingzhi ( 史 [UNK] [UNK] , shih ping - chih )' in
'was Shi Bingzhi (史秉直, Shih Ping-chih).'
I0825 16:32:22.661446 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'paper banknotes ( [UNK] , chao ) . pax mongolica , mongol
peace' in 'paper banknotes (鈔, Chao). Pax Mongolica, Mongol peace,'
I0825 16:32:22.663194 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: '[UNK] , chao ) . pax mongolica , mongol peace' in '(鈔, Cha
o). Pax Mongolica, Mongol peace,'
I0825 16:32:22.793176 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: '大 元 [UNK] [UNK] , " the comprehensive institutions of the
```

```
great yuan' in '大元通制, "the comprehensive institutions of the Great Yua
n"),'
I0825 16:32:22.795842 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the da yuan tong zhi ( chinese : 大 元 [UNK] [UNK] , " the
comprehensive institutions of the great yuan' in 'the Da Yuan Tong Zhi (Chine
se: 大元通制, "the comprehensive institutions of the Great Yuan"),'
I0825 16:32:22.798540 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'chinese : 大 元 [UNK] [UNK] , " the comprehensive instituti
ons of the great yuan' in '(Chinese: 大元通制, "the comprehensive institutions
of the Great Yuan"),'
I0825 16:32:22.801251 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'da yuan tong zhi ( chinese : 大 元 [UNK] [UNK] , " the comp
rehensive institutions of the great yuan' in 'Da Yuan Tong Zhi (Chinese: 大元
通制, "the comprehensive institutions of the Great Yuan"),'
I0825 16:32:22.803255 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: '( chinese : 大 元 [UNK] [UNK] , " the comprehensive institu
tions of the great yuan' in '(Chinese: 大元通制, "the comprehensive institutio
ns of the Great Yuan"),'
I0825 16:32:22.841369 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'academy of the pavilion of the star of literature ( chines
e : [UNK] 章 [UNK] [UNK] 士 [UNK] )' in 'Academy of the Pavilion of the Star o
f Literature (Chinese: 奎章閣學士院),'
I0825 16:32:22.843837 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the pavilion of the star of literature ( chinese : [UNK] 章
[UNK] [UNK] 士 [UNK] )' in 'the Pavilion of the Star of Literature (Chinese:
奎章閣學士院),'
I0825 16:32:22.845887 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the academy of the pavilion of the star of literature ( ch
inese : [UNK] 章 [UNK] [UNK] 士 [UNK] )' in 'the Academy of the Pavilion of th
e Star of Literature (Chinese: 奎章閣學士院),'
I0825 16:32:22.847403 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'academy of the pavilion of the star of literature ( chines
e : [UNK] 章 [UNK] [UNK] 士 [UNK] ) ,' in 'Academy of the Pavilion of the Star
of Literature (Chinese: 奎章閣學士院),'
I0825 16:32:22.849205 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'pavilion of the star of literature ( chinese : [UNK] 章 [UN
K] [UNK] 士 [UNK] )' in 'Pavilion of the Star of Literature (Chinese: 奎章閣學
士院),'
I0825 16:32:22.851174 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the pavilion of the star of literature ( chinese : [UNK] 章
[UNK] [UNK] 士 [UNK] ) ,' in 'the Pavilion of the Star of Literature (Chinese:
奎章閣學士院),'
I0825 16:32:22.852656 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the academy of the pavilion of the star of literature ( ch
inese : [UNK] 章 [UNK] [UNK] 士 [UNK] ) ,' in 'the Academy of the Pavilion of
the Star of Literature (Chinese: 奎章閣學士院),'
I0825 16:32:22.853934 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'star of literature ( chinese : [UNK] 章 [UNK] [UNK] 士 [UN
K] )' in 'Star of Literature (Chinese: 奎章閣學士院),'
I0825 16:32:22.858396 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'founding the academy of the pavilion of the star of litera
ture ( chinese : [UNK] 章 [UNK] [UNK] 士 [UNK] ) , first established in the sp
ring of 1329' in 'founding the Academy of the Pavilion of the Star of Literat
ure (Chinese: 奎章閣學士院), first established in the spring of 1329'
I0825 16:32:22.860411 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'academy of the pavilion of the star of literature ( chines
e : [UNK] 章 [UNK] [UNK] 士 [UNK] ) , first established in the spring of 1329'
```

```
in 'Academy of the Pavilion of the Star of Literature (Chinese: 奎章閣學士院),
first established in the spring of 1329'
I0825 16:32:22.861855 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: '士 [UNK] ) , first established in the spring of 1329' in
'奎章閣學士院), first established in the spring of 1329'
I0825 16:32:22.864022 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the academy of the pavilion of the star of literature ( ch
inese : [UNK] 章 [UNK] [UNK] 士 [UNK] ) , first established in the spring of 1
329' in 'the Academy of the Pavilion of the Star of Literature (Chinese: 奎章
閣學士院), first established in the spring of 1329'
I0825 16:32:22.865864 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'chinese : [UNK] 章 [UNK] [UNK] 士 [UNK] ) , first establish
ed in the spring of 1329' in '(Chinese: 奎章閣學士院), first established in th
e spring of 1329'
I0825 16:32:22.867182 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: '[UNK] 章 [UNK] [UNK] 士 [UNK] ) , first established in the
spring of 1329' in '奎章閣學士院), first established in the spring of 1329'
I0825 16:32:22.873031 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'jingshi dadian ( chinese : [UNK] 世 大 [UNK] )' in 'Jingshi
Dadian (Chinese: 經世大典).'
I0825 16:32:22.874607 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'jingshi dadian ( chinese : [UNK] 世 大 [UNK] ) .' in 'Jings
hi Dadian (Chinese: 經世大典).'
I0825 16:32:22.877229 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'jingshi dadian ( chinese : [UNK] 世 大 [UNK]' in 'Jingshi D
adian (Chinese: 經世大典).'
I0825 16:32:23.096546 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'central secretariat ( zhongshu sheng ) to manage civil aff
airs , the privy council ( chinese : [UNK] [UNK] [UNK] ) to manage military a
ffairs' in 'Central Secretariat (Zhongshu Sheng) to manage civil affairs, the
Privy Council (Chinese: 樞密院) to manage military affairs,'
I0825 16:32:23.099749 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'manage civil affairs , the privy council ( chinese : [UNK]
[UNK] [UNK] ) to manage military affairs , and the censorate to conduct inter
nal surveillance and inspection' in 'manage civil affairs, the Privy Council
(Chinese: 樞密院) to manage military affairs, and the Censorate to conduct int
ernal surveillance and inspection.'
I0825 16:32:23.232178 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK] [UNK] )' in 'Shoushi Li (授時曆)'
I0825 16:32:23.233923 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK] [UNK] ) or calendar for fixing th
e seasons' in 'Shoushi Li (授時曆) or Calendar for Fixing the Seasons,'
I0825 16:32:23.235562 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK] [UNK] ) or calendar for fixing th
e seasons ,' in 'Shoushi Li (授時曆) or Calendar for Fixing the Seasons,'
I0825 16:32:23.237160 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK] [UNK]' in 'Shoushi Li (授時曆)'
I0825 16:32:23.238773 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the shoushi li ( [UNK] [UNK] [UNK] )' in 'the Shoushi Li
(授時曆)'
I0825 16:32:23.241008 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the shoushi li ( [UNK] [UNK] [UNK] ) or calendar for fixin
g the seasons' in 'the Shoushi Li (授時曆) or Calendar for Fixing the Season
s,'
I0825 16:32:23.242423 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK] [UNK] ) or calendar' in 'Shoushi
Li (授時曆) or Calendar'
```

```
I0825 16:32:23.243585 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK]' in 'Shoushi Li (授時曆)'
I0825 16:32:23.244728 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK]' in 'Shoushi Li (授時曆)'
I0825 16:32:23.246573 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK] [UNK] ) or' in 'Shoushi Li (授時
曆) or'
I0825 16:32:23.247848 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the shoushi li ( [UNK] [UNK] [UNK] ) or calendar for fixin
g the seasons ,' in 'the Shoushi Li (授時曆) or Calendar for Fixing the Season
s,'
I0825 16:32:23.251555 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK] [UNK] ) or calendar for fixing th
e seasons' in 'Shoushi Li (授時曆) or Calendar for Fixing the Seasons,'
I0825 16:32:23.252835 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK] [UNK] )' in 'Shoushi Li (授時曆)'
I0825 16:32:23.254248 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK] [UNK] ) or calendar for fixing th
e seasons ,' in 'Shoushi Li (授時曆) or Calendar for Fixing the Seasons,'
I0825 16:32:23.255840 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK] [UNK]' in 'Shoushi Li (授時曆)'
I0825 16:32:23.257169 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK] [UNK] ) or calendar' in 'Shoushi
Li (授時曆) or Calendar'
I0825 16:32:23.258476 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the shoushi li ( [UNK] [UNK] [UNK] ) or calendar for fixin
g the seasons' in 'the Shoushi Li (授時曆) or Calendar for Fixing the Season
s,'
I0825 16:32:23.259843 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the shoushi li ( [UNK] [UNK] [UNK] )' in 'the Shoushi Li
(授時曆)'
I0825 16:32:23.261180 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK]' in 'Shoushi Li (授時曆)'
I0825 16:32:23.262257 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK]' in 'Shoushi Li (授時曆)'
I0825 16:32:23.263583 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK] [UNK] ) or calendar for fixing th
e seasons , was disseminated in 1281' in 'Shoushi Li (授時曆) or Calendar for
Fixing the Seasons, was disseminated in 1281'
I0825 16:32:23.264923 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the shoushi li ( [UNK] [UNK] [UNK] ) or calendar for fixin
g the seasons ,' in 'the Shoushi Li (授時曆) or Calendar for Fixing the Season
s,'
I0825 16:32:23.266135 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK] [UNK] ) or' in 'Shoushi Li (授時
曆) or'
I0825 16:32:23.270689 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'shoushi li ( [UNK] [UNK] [UNK] ) or calendar for fixing th
e seasons , was disseminated in 1281' in 'Shoushi Li (授時曆) or Calendar for
Fixing the Seasons, was disseminated in 1281'
I0825 16:32:23.272567 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'his calendar , the shoushi li ( [UNK] [UNK] [UNK] ) or cal
endar for fixing the seasons , was disseminated in 1281' in 'His calendar, th
e Shoushi Li (授時曆) or Calendar for Fixing the Seasons, was disseminated in
1281'
I0825 16:32:23.274293 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'the shoushi li ( [UNK] [UNK] [UNK] ) or calendar for fixin
```

```
g the seasons , was disseminated in 1281' in 'the Shoushi Li (授時暦) or Calen
dar for Fixing the Seasons, was disseminated in 1281'
I0825 16:32:23.276243 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'calendar , the shoushi li ( [UNK] [UNK] [UNK] ) or calenda
r for fixing the seasons , was disseminated in 1281' in 'calendar, the Shoush
i Li (授時暦) or Calendar for Fixing the Seasons, was disseminated in 1281'
I0825 16:32:23.277883 140319166060416 <ipython-input-12-12e5f9aec9d7>:27] Una
ble to find text: 'li ( [UNK] [UNK] [UNK] ) or calendar for fixing the season
s , was disseminated in 1281' in 'Li (授時暦) or Calendar for Fixing the Seaso
ns, was disseminated in 1281'
I0825 16:32:27.223864 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Swiss-Austrianborder.Itislocatedatappr
oximately47°39′N9°19′E' vs 'swiss-austrianborder.itislocatedatapproximately4
7°39′n9°19′e'
I0825 16:32:29.016467 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Islamism,alsoknownasPoliticalIslam(Ara
bic:إسلامسياسيislāmsiyāsī),' vs 'islamism,alsoknownaspoliticalislam(arabic:لامسياسي
islamsiyasi),'
I0825 16:32:29.018499 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Islamism,alsoknownasPoliticalIslam(Ara
bic:إسلامسياسيislāmsiyāsī),' vs 'islamism,alsoknownaspoliticalislam(arabic:لامسياسي
islamsiyasi),'
I0825 16:32:29.020376 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Islamism,alsoknownasPoliticalIslam(Ara
bic:إسلامسياسيislāmsiyāsī),' vs 'islamism,alsoknownaspoliticalislam(arabic:لامسياسي
islamsiyasi),'
I0825 16:32:29.022231 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Islamism,alsoknownasPoliticalIslam(Ara
bic:إسلامسياسي' vs 'islamism,alsoknownaspoliticalislam(arabic:اسلامسياسي'
I0825 16:32:29.023923 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Islamism,alsoknownasPoliticalIslam(Ara
bic:إسلامسياسيislāmsiyāsī),isanIslamic' vs 'islamism,alsoknownaspoliticalislam(ar
abic:اسلامسياسيislamsiyasi),isanislamic'
I0825 16:32:29.025262 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'PoliticalIslam(Arabic:إسلامسياسيislāmsiyās
ī),' vs 'politicalislam(arabic:اسلامسياسيislamsiyasi),'
I0825 16:32:29.026864 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'PoliticalIslam(Arabic:إسلامسياسيislāmsiyās
ī),' vs 'politicalislam(arabic:اسلامسياسيislamsiyasi),'
I0825 16:32:29.028142 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'PoliticalIslam(Arabic:إسلامسياسيislāmsiyās
ī),' vs 'politicalislam(arabic:اسلامسياسيislamsiyasi),'
I0825 16:32:29.029619 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'PoliticalIslam(Arabic:إسلامسياسيislāmsiyās
ī),isanIslamicrevivalmovement' vs 'politicalislam(arabic:اسلامسياسيislamsiyasi),i
sanislamicrevivalmovement'
I0825 16:32:29.031355 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'PoliticalIslam(Arabic:إسلامسياسيislāmsiyās
ī),isanIslamicrevival' vs 'politicalislam(arabic:اسلامسياسيislamsiyasi),isanislam
icrevival'
I0825 16:32:29.032604 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'PoliticalIslam(Arabic:إسلامسياسي' vs 'poli
ticalislam(arabic:اسلامسياسي'
I0825 16:32:29.033915 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'PoliticalIslam(Arabic:إسلامسياسيislāmsiyās
ī),isanIslamic' vs 'politicalislam(arabic:اسلامسياسيislamsiyasi),isanislamic'
I0825 16:32:29.035189 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
```

```
gth not equal after stripping spaces: 'Islam(Arabic:إسلامسياسيislāmsiyāsī),' vs
'islam(arabic:اسلامسياسيislamsiyasi),'
I0825 16:32:29.036619 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Islamism,alsoknownasPoliticalIslam(Ara
bic:إسلامسياسيislāmsiyāsī),' vs 'islamism,alsoknownaspoliticalislam(arabic:لامسياسي
islamsiyasi),'
I0825 16:32:29.037911 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Islam(Arabic:إسلامسياسيislāmsiyāsī),' vs
'islam(arabic:اسلامسياسيislamsiyasi),'
I0825 16:32:29.039280 140319166060416 <ipython-input-12-12e5f9aec9d7>:37] Len
gth not equal after stripping spaces: 'Islamism,alsoknownasPoliticalIslam(Ara
bic:إسلامسياسيislāmsiyāsī),' vs 'islamism,alsoknownaspoliticalislam(arabic:لامسياسي
islamsiyasi),'
```

F1 Score Calcualtion

In [0]:
```
# please refer https://github.com/allenai/bi-att-flow/blob/master/squad/evaluate-v1.1.py#L86
import re
from collections import Counter
def normalize_answer(s):
    """Lower text and remove punctuation, articles and extra whitespace."""
    def remove_articles(text):
        return re.sub(r'\b(a|an|the)\b', ' ', text)

    def white_space_fix(text):
        return ' '.join(text.split())

    def remove_punc(text):
        exclude = set(string.punctuation)
        return ''.join(ch for ch in text if ch not in exclude)

    def lower(text):
        return text.lower()

    return white_space_fix(remove_articles(remove_punc(lower(s))))

def f1_score(prediction, ground_truth):
    prediction_tokens = normalize_answer(prediction).split()
    ground_truth_tokens = normalize_answer(ground_truth).split()
    common = Counter(prediction_tokens) & Counter(ground_truth_tokens)
    num_same = sum(common.values())
    if num_same == 0:
        return 0
    precision = 1.0 * num_same / len(prediction_tokens)
    recall = 1.0 * num_same / len(ground_truth_tokens)
    f1 = (2 * precision * recall) / (precision + recall)
    return f1


def exact_match_score(prediction, ground_truth):
    return (normalize_answer(prediction) == normalize_answer(ground_truth))


def metric_max_over_ground_truths(metric_fn, prediction, ground_truths):
    scores_for_ground_truths = []
    for ground_truth in ground_truths:
        score = metric_fn(prediction, ground_truth)
        scores_for_ground_truths.append(score)
    return max(scores_for_ground_truths)


def evaluate(dataset, predictions):
    f1 = exact_match = total = 0
    for article in dataset:
        for paragraph in article['paragraphs']:
            for qa in paragraph['qas']:
                total += 1
                if qa['id'] not in predictions:
#                     message = 'Unanswered question ' + qa['id'] + \
#                               ' will receive score 0.'
#                     print(message, file=sys.stderr)
```

```
                    continue
                ground_truths = list(map(lambda x: x['text'], qa['answers']))
                prediction = predictions[qa['id']]
                exact_match += metric_max_over_ground_truths(
                    exact_match_score, prediction, ground_truths)
                f1 += metric_max_over_ground_truths(
                    f1_score, prediction, ground_truths)

    exact_match = 100.0 * exact_match / total
    f1 = 100.0 * f1 / total

    return {'exact_match': exact_match, 'f1': f1}


def evaluate_squad(data_file,pred_file):

  with open(data_file) as dataset_file:
    dataset_json = json.load(dataset_file)
    dataset = dataset_json['data']
  with open(pred_file) as prediction_file:
    predictions = json.load(prediction_file)
  print(evaluate(dataset, predictions))
```

In [44]:
```
evaluate_squad("/content/dev-v1.1.json","/content/bert-checkpoints_models_SQUAD_predictions.json")
```

```
{'exact_match': 80.85146641438033, 'f1': 88.0228956599229}
```

In [48]:
```
evaluate_squad("/content/train-v1.1.json","/content/bert-checkpoints_models_SQUAD_train_predictions.json")
```

```
{'exact_match': 84.38978240302744,'f1': 90.87081895814865}
```