NNproject(squadv2) (1)

April 23, 2023

[ ]:
```
pip install transformers datasets evaluate
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting transformers
  Downloading transformers-4.28.1-py3-none-any.whl (7.0 MB)
                            7.0/7.0 MB
40.6 MB/s eta 0:00:00
Collecting datasets
  Downloading datasets-2.11.0-py3-none-any.whl (468 kB)
                           468.7/468.7 kB
25.1 MB/s eta 0:00:00
Collecting evaluate
  Downloading evaluate-0.4.0-py3-none-any.whl (81 kB)
                            81.4/81.4 kB
5.6 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.17 in
/usr/local/lib/python3.9/dist-packages (from transformers) (1.22.4)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.9/dist-
packages (from transformers) (4.65.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.9/dist-
packages (from transformers) (3.11.0)
Collecting huggingface-hub<1.0,>=0.11.0
  Downloading huggingface_hub-0.13.4-py3-none-any.whl (200 kB)
                           200.1/200.1 kB
14.9 MB/s eta 0:00:00
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.9/dist-packages (from transformers) (2022.10.31)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.9/dist-
packages (from transformers) (6.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-
packages (from transformers) (23.1)
Collecting tokenizers!=0.11.3,<0.14,>=0.11.1
  Downloading
tokenizers-0.13.3-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.8
MB)
                            7.8/7.8 MB
```

```
62.2 MB/s eta 0:00:00
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-
packages (from transformers) (2.27.1)
Requirement already satisfied: pyarrow>=8.0.0 in /usr/local/lib/python3.9/dist-
packages (from datasets) (9.0.0)
Requirement already satisfied: fsspec[http]>=2021.11.1 in
/usr/local/lib/python3.9/dist-packages (from datasets) (2023.4.0)
Collecting multiprocess
  Downloading multiprocess-0.70.14-py39-none-any.whl (132 kB)
                             132.9/132.9
kB 8.2 MB/s eta 0:00:00
Collecting responses<0.19
  Downloading responses-0.18.0-py3-none-any.whl (38 kB)
Collecting xxhash
  Downloading
xxhash-3.2.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (212 kB)
                             212.2/212.2
kB 9.4 MB/s eta 0:00:00
Collecting aiohttp
  Downloading
aiohttp-3.8.4-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.0 MB)
                             1.0/1.0 MB
17.7 MB/s eta 0:00:00
Collecting dill<0.3.7,>=0.3.0
  Downloading dill-0.3.6-py3-none-any.whl (110 kB)
                             110.5/110.5
kB 7.6 MB/s eta 0:00:00
Requirement already satisfied: pandas in /usr/local/lib/python3.9/dist-
packages (from datasets) (1.5.3)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.9/dist-
packages (from aiohttp->datasets) (23.1.0)
Collecting aiosignal>=1.1.2
  Downloading aiosignal-1.3.1-py3-none-any.whl (7.6 kB)
Collecting frozenlist>=1.1.1
  Downloading frozenlist-1.3.3-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x86_64.
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (158 kB)
                             158.8/158.8 kB
11.2 MB/s eta 0:00:00
Collecting yarl<2.0,>=1.0
  Downloading
yarl-1.9.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (269 kB)
                             269.3/269.3 kB
12.4 MB/s eta 0:00:00
Collecting async-timeout<5.0,>=4.0.0a3
  Downloading async_timeout-4.0.2-py3-none-any.whl (5.8 kB)
Collecting multidict<7.0,>=4.5
```

```
  Downloading
multidict-6.0.4-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (114
kB)
                            114.2/114.2
kB 4.5 MB/s eta 0:00:00
Requirement already satisfied: charset-normalizer<4.0,>=2.0 in
/usr/local/lib/python3.9/dist-packages (from aiohttp->datasets) (2.0.12)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.9/dist-packages (from huggingface-
hub<1.0,>=0.11.0->transformers) (4.5.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-
packages (from requests->transformers) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.9/dist-packages (from requests->transformers) (2022.12.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.9/dist-packages (from requests->transformers) (1.26.15)
Requirement already satisfied: python-dateutil>=2.8.1 in
/usr/local/lib/python3.9/dist-packages (from pandas->datasets) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.9/dist-
packages (from pandas->datasets) (2022.7.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-
packages (from python-dateutil>=2.8.1->pandas->datasets) (1.16.0)
Installing collected packages: tokenizers, xxhash, multidict, frozenlist, dill,
async-timeout, yarl, responses, multiprocess, huggingface-hub, aiosignal,
transformers, aiohttp, datasets, evaluate
Successfully installed aiohttp-3.8.4 aiosignal-1.3.1 async-timeout-4.0.2
datasets-2.11.0 dill-0.3.6 evaluate-0.4.0 frozenlist-1.3.3 huggingface-
hub-0.13.4 multidict-6.0.4 multiprocess-0.70.14 responses-0.18.0
tokenizers-0.13.3 transformers-4.28.1 xxhash-3.2.0 yarl-1.9.1
```

[ ]:
```python
from huggingface_hub import notebook_login

notebook_login()
```

```
Token is valid.
Your token has been saved in your configured git credential helpers (store).
Your token has been saved to /root/.cache/huggingface/token
Login successful
```

[ ]:
```python
from datasets import load_dataset

squad = load_dataset("squad_v2", split="train[:1000]")

squad = squad.train_test_split(test_size=0.2)

squad["train"][0]
```

```
{'answers': {'answer_start': [515], 'text': ['Saint Bernadette Soubirous']},
```

```
 'context': 'Architecturally, the school has a Catholic 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 upraised with the legend "Venite Ad Me Omnes". Next to␣
 ↪the Main Building is the Basilica of the Sacred Heart. Immediately behind␣
 ↪the basilica is the Grotto, a Marian place of prayer and reflection. It is a␣
 ↪replica of the grotto at Lourdes, France where the Virgin Mary reputedly␣
 ↪appeared to Saint Bernadette Soubirous in 1858. At the end of the main drive␣
 ↪(and in a direct line that connects through 3 statues and the Gold Dome), is␣
 ↪a simple, modern stone statue of Mary.',
 'id': '5733be284776f41900661182',
 'question': 'To whom did the Virgin Mary allegedly appear in 1858 in Lourdes␣
 ↪France?',
 'title': 'University_of_Notre_Dame'
}
```

```
Downloading builder script:   0%|          | 0.00/5.28k [00:00<?, ?B/s]
```

```
Downloading metadata:   0%|          | 0.00/2.40k [00:00<?, ?B/s]
```

```
Downloading readme:   0%|          | 0.00/8.02k [00:00<?, ?B/s]
```

```
Downloading and preparing dataset squad_v2/squad_v2 to /root/.cache/huggingface/
datasets/squad_v2/squad_v2/2.0.0/09187c73c1b837c95d9a249cd97c2c3f1cebada06efe667
b4427714b27639b1d…
```

```
Downloading data files:   0%|          | 0/2 [00:00<?, ?it/s]
```

```
Downloading data:   0%|          | 0.00/9.55M [00:00<?, ?B/s]
```

```
Downloading data:   0%|          | 0.00/801k [00:00<?, ?B/s]
```

```
Extracting data files:   0%|          | 0/2 [00:00<?, ?it/s]
```

```
Generating train split:   0%|          | 0/130319 [00:00<?, ? examples/s]
```

```
Generating validation split:   0%|          | 0/11873 [00:00<?, ? examples/s]
```

```
Dataset squad_v2 downloaded and prepared to /root/.cache/huggingface/datasets/sq
uad_v2/squad_v2/2.0.0/09187c73c1b837c95d9a249cd97c2c3f1cebada06efe667b4427714b27
639b1d. Subsequent calls will reuse this data.
```

[ ]:
```
{'answers': {'answer_start': [515], 'text': ['Saint Bernadette Soubirous']},
 'context': 'Architecturally, the school has a Catholic 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
upraised with the legend "Venite Ad Me Omnes". Next to the Main Building is the
Basilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a
Marian place of prayer and reflection. It is a replica of the grotto at Lourdes,
France where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in
1858. At the end of the main drive (and in a direct line that connects through 3
statues and the Gold Dome), is a simple, modern stone statue of Mary.',
```

```
 'id': '5733be284776f41900661182',
 'question': 'To whom did the Virgin Mary allegedly appear in 1858 in Lourdes
France?',
 'title': 'University_of_Notre_Dame'}
```

[ ]:
```
# from transformers import AutoTokenizer

# tokenizer = AutoTokenizer.from_pretrained("deepset/roberta-base-squad2")

# def preprocess_function(examples):
#     questions = [q.strip() for q in examples["question"]]
#     inputs = tokenizer(
#         questions,
#         examples["context"],
#         max_length=384,
#         truncation="only_second",
#         return_offsets_mapping=True,
#         padding="max_length",
#     )

#     offset_mapping = inputs.pop("offset_mapping")
#     answers = examples["answers"]
#     start_positions = []
#     end_positions = []

#     for i, offset in enumerate(offset_mapping):
#         answer = answers[i]
#         if len(answer["answer_start"]) == 0:
#             start_positions.append(0)
#             end_positions.append(0)
#         else:
#             start_char = answer["answer_start"][0]
#             end_char = answer["answer_start"][0] + len(answer["text"][0])
#             sequence_ids = inputs.sequence_ids(i)

#             # Find the start and end of the context
#             idx = 0
#             while sequence_ids[idx] != 1:
#                 idx += 1
#             context_start = idx
#             while sequence_ids[idx] == 1:
#                 idx += 1
#             context_end = idx - 1

#             # If the answer is not fully inside the context, label it (0, 0)
#             if offset[context_start][0] > end_char or offset[context_end][1]
 ↪< start_char:
```

```
#                 start_positions.append(0)
#                 end_positions.append(0)
#             else:
#                 # Otherwise it's the start and end token positions
#                 idx = context_start
#                 while idx <= context_end and offset[idx][0] <= start_char:
#                     idx += 1
#                 start_positions.append(idx - 1)

#                 idx = context_end
#                 while idx >= context_start and offset[idx][1] >= end_char:
#                     idx -= 1
#                 end_positions.append(idx + 1)

#     inputs["start_positions"] = start_positions
#     inputs["end_positions"] = end_positions
#     return inputs


from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("deepset/roberta-base-squad2")

def preprocess_function(examples):
    questions = [q.strip() for q in examples["question"]]
    inputs = tokenizer(
        questions,
        examples["context"],
        max_length=384,
        truncation="only_second",
        return_offsets_mapping=True,
        padding="max_length",
    )

    offset_mapping = inputs.pop("offset_mapping")
    answers = examples["answers"]
    start_positions = []
    end_positions = []

    for i, offset in enumerate(offset_mapping):
        answer = answers[i]
        if len(answer["answer_start"]) == 0:
            start_positions.append(-1)  # Set to -1 if no answer can be found
            end_positions.append(-1)  # Set to -1 if no answer can be found
        else:
            start_char = answer["answer_start"][0]
            end_char = answer["answer_start"][0] + len(answer["text"][0])
```

```
            sequence_ids = inputs.sequence_ids(i)

            # Find the start and end of the context
            idx = 0
            while sequence_ids[idx] != 1:
                idx += 1
            context_start = idx
            while sequence_ids[idx] == 1:
                idx += 1
            context_end = idx - 1

            # If the answer is not fully inside the context, label it (-1, -1)
            if offset[context_start][0] > end_char or offset[context_end][1] <␣
 ↪start_char:
                start_positions.append(0)
                end_positions.append(0)
            else:
                # Otherwise it's the start and end token positions
                idx = context_start
                while idx <= context_end and offset[idx][0] <= start_char:
                    idx += 1
                start_positions.append(idx - 1)

                idx = context_end
                while idx >= context_start and offset[idx][1] >= end_char:
                    idx -= 1
                end_positions.append(idx + 1)

    inputs["start_positions"] = start_positions
    inputs["end_positions"] = end_positions
    return inputs
```

[ ]:
```
tokenized_squad = squad.map(preprocess_function, batched=True,␣
 ↪remove_columns=squad["train"].column_names)
```

```
Map:   0%|          | 0/800 [00:00<?, ? examples/s]
Map:   0%|          | 0/200 [00:00<?, ? examples/s]
```

[ ]:
```
from transformers import DefaultDataCollator

data_collator = DefaultDataCollator()
```

[ ]:
```
from transformers import AutoModelForQuestionAnswering, TrainingArguments,␣
 ↪Trainer
```

```
model = AutoModelForQuestionAnswering.from_pretrained("deepset/
 ↪tinyroberta-squad2")

training_args = TrainingArguments(
    output_dir="my_awesome_qa_model",
    evaluation_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=3,
    weight_decay=0.01,
    push_to_hub=True,
)

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_squad["train"],
    eval_dataset=tokenized_squad["test"],
    tokenizer=tokenizer,
    data_collator=data_collator,
)

trainer.train()
```

```
Downloading (…)lve/main/config.json:   0%|          | 0.00/835 [00:00<?, ?B/s]

Downloading pytorch_model.bin:   0%|          | 0.00/326M [00:00<?, ?B/s]

Cloning https://huggingface.co/Rekhni/my_awesome_qa_model into local empty
directory.
WARNING:huggingface_hub.repository:Cloning
https://huggingface.co/Rekhni/my_awesome_qa_model into local empty directory.

Download file pytorch_model.bin:   0%|          | 8.00k/415M [00:00<?, ?B/s]

Download file training_args.bin: 100%|##########| 3.50k/3.50k [00:00<?, ?B/s]

Download file runs/Apr20_04-15-21_24ce5b36a57e/1681964126.999821/events.out.
 ↪tfevents.1681964126.24ce5b36a57e.4…

Download file runs/Apr20_22-49-53_bba4958eb40b/events.out.tfevents.1682031069.
 ↪bba4958eb40b.1400.0: 100%|######…

Clean file training_args.bin:  29%|##8       | 1.00k/3.50k [00:00<?, ?B/s]

Download file runs/Apr20_04-15-21_24ce5b36a57e/events.out.tfevents.1681964126.
 ↪24ce5b36a57e.472.0: 100%|#######…

Clean file runs/Apr20_04-15-21_24ce5b36a57e/1681964126.999821/events.out.
 ↪tfevents.1681964126.24ce5b36a57e.472.…
```

```
Download file runs/Apr20_22-49-53_bba4958eb40b/1682031069.8927307/events.out.
  ↪tfevents.1682031069.bba4958eb40b.…

Clean file runs/Apr20_22-49-53_bba4958eb40b/events.out.tfevents.1682031069.
  ↪bba4958eb40b.1400.0:  23%|##3      …

Clean file runs/Apr20_04-15-21_24ce5b36a57e/events.out.tfevents.1681964126.
  ↪24ce5b36a57e.472.0:  20%|#9        …

Clean file runs/Apr20_22-49-53_bba4958eb40b/1682031069.8927307/events.out.
  ↪tfevents.1682031069.bba4958eb40b.140…

Clean file pytorch_model.bin:   0%|          | 1.00k/415M [00:00<?, ?B/s]

/usr/local/lib/python3.9/dist-packages/transformers/optimization.py:391:
FutureWarning: This implementation of AdamW is deprecated and will be removed in
a future version. Use the PyTorch implementation torch.optim.AdamW instead, or
set `no_deprecation_warning=True` to disable this warning
  warnings.warn(

<IPython.core.display.HTML object>
```

[ ]:
```
TrainOutput(global_step=150, training_loss=0.3517026519775391,
metrics={'train_runtime': 5715.8327, 'train_samples_per_second': 0.42,
'train_steps_per_second': 0.026, 'total_flos': 235175580057600.0, 'train_loss':
0.3517026519775391, 'epoch': 3.0})
```

[ ]:
```
from transformers import pipeline
question_answerer = pipeline("question-answering", model="my_awesome_qa_model")

context = """Computational complexity theory is a branch of the theory of␣
 ↪computation in theoretical computer science\
that focuses on classifying computational problems according to their inherent␣
 ↪difficulty, and relating\
those classes to each other. A computational problem is understood to be a task␣
 ↪that is in principle\
amenable to being solved by a computer, which is equivalent to stating that the␣
 ↪problem may be solved\
by mechanical application of mathematical steps, such as an algorithm.
"""
question1 = """What branch of theoretical computer science deals with broadly␣
 ↪classifying computational problems by difficulty and class of relationship?
 ↪"""
question2 = """By what main attribute are computational problems classified␣
 ↪utilizing computational complexity theory?"""
question3 = """What is the term for a task that generally lends itself to being␣
 ↪solved by a computer?"""
question4 = """What is computational complexity principle?"""
```

```
question5 = """What branch of theoretical computer class deals with broadly␣
 ↪classifying computational problems by difficulty and class of relationship?
 ↪"""
question6 = """What is understood to be a task that is in principle not␣
 ↪amendable to being solved by a computer?"""
question7 = """What cannot be solved by mechanical application of mathematical␣
 ↪steps?"""
question8 = """What is a manual application of mathematical steps?"""

questions = [question1, question2, question3, question4, question5, question6,␣
 ↪question7, question8]



for q in questions:
    answer = question_answerer(question=q, context=context)
    if answer["score"] < 0.2:
        print("Question: ", q)
        print("Answer: <No Answer>\n")
    else:
        print("Question: ", q)
        print("Answer: ", answer["answer"], "\n")
```

```
Question:  What branch of theoretical computer science deals with broadly
classifying computational problems by difficulty and class of relationship?
Answer:  Computational complexity theory

Question:  By what main attribute are computational problems classified
utilizing computational complexity theory?
Answer: <No Answer>

Question:  What is the term for a task that generally lends itself to being
solved by a computer?
Answer:  mechanical application of mathematical steps, such as an algorithm

Question:  What is computational complexity principle?
Answer: <No Answer>

Question:  What branch of theoretical computer class deals with broadly
classifying computational problems by difficulty and class of relationship?
Answer:  Computational complexity theory

Question:  What is understood to be a task that is in principle not amendable to
being solved by a computer?
Answer: <No Answer>

Question:  What cannot be solved by mechanical application of mathematical
```

steps?
Answer: <No Answer>

Question: What is a manual application of mathematical steps?
Answer: mechanical application of mathematical steps, such as an algorithm