

```
from google.colab import drive
drive.mount("/content/drive", force_remount=True)

Mounted at /content/drive

google_drive_path = '/content/drive/MyDrive/XAI'
```

## Tweet Sentiment Extraction - Hugging Face QA Model

This notebook tries to implement Hugging Face Question Answering model to extract text in the input text that results in the sentiment label. At the end of the notebook, we also apply SHAP Question Answering explainer to explain the behavior of the model.

The model uses the pretrained *distilbert-base-uncased* AutoTokenizer and AutoModelForQuestionAnswering.

### Load Libraries, Data

We will use HuggingFace transformers [here](#)

```
import pandas as pd, numpy as np
import tensorflow as tf
import tensorflow.keras.backend as K
from sklearn.model_selection import StratifiedKFold
from transformers import *
import tokenizers

print('TF version',tf.__version__)

/usr/local/lib/python3.10/dist-packages/transformers/deepspeed.py:23: FutureWarning: transformers.deepspeed module is deprecated and will be removed in a fu
warnings.warn(
WARNING:jax._src.xla_bridge:CUDA backend failed to initialize: Found cuBLAS version 120103, but JAX was built against version 120205, which is newer. The co
/usr/local/lib/python3.10/dist-packages/transformers/generation_utils.py:24: FutureWarning: Importing `GenerationMixin` from `src/transformers/generation_ut
warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation_tf_utils.py:24: FutureWarning: Importing `TFGenerationMixin` from `src/transformers/generati
warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation_flax_utils.py:24: FutureWarning: Importing `FlaxGenerationMixin` from `src/transformers/gene
warnings.warn(
TF version 2.15.0
```

```
train = pd.read_csv(google_drive_path+'input/tweet-sentiment-extraction/train.csv').fillna('')
test = pd.read_csv(google_drive_path+'input/tweet-sentiment-extraction/test.csv').fillna('')
train.head()
```

	textID	text	selected_text	sentiment
0	cb774db0d1	I'd have responded, if I were going	I'd have responded, if I were going	neutral
1	549e992a42	Sooo SAD I will miss you here in San Diego!!!	Sooo SAD	negative
2	088c60f138	my boss is bullying me...	bullying me	negative
3	9642c003ef	what interview! leave me alone	leave me alone	negative
4	358bd9e861	Sons of ****, why couldn't they put them on t...	Sons of ****,	negative

### Prepare dataset for implementing Question Answering Transformers Model

We will now add two columns to the data frame, which is needed when training a QA model.

```
train_qa = train.copy()
test_qa = test.copy()

# Add column question and answer_start to the dataset
train_qa['question'] = 'Why is this sentiment '+train_qa['sentiment']+'?'
test_qa['question'] = 'Why is this sentiment '+test_qa['sentiment']+'?'

train_qa['answer_start'] = np.nan
```

```

for k in range(train_qa.shape[0]):
    text1 = " "+" ".join(train_qa.loc[k, 'text'].split())
    text2 = " "+" ".join(train_qa.loc[k, 'selected_text'].split())
    idx = text1.find(text2)
    idx = idx - 1
    train_qa.loc[k, 'answer_start'] = idx

```

```

train_qa['answer_start'] = train_qa['answer_start'].astype(int)
train_qa.head()

```

	textID	text	selected_text	sentiment	question	answer_start
0	cb774db0d1	I'd have responded, if I were going	I'd have responded, if I were going	neutral	Why is this sentiment neutral?	0
1	549e992a42	Sooo SAD I will miss you here in San Diego!!!	Sooo SAD	negative	Why is this sentiment negative?	0
2	088c60f138	my boss is bullying me...	bullying me	negative	Why is this sentiment negative?	11
3	881e33e3f	what interview! leave me	.	..	Why is this sentiment	10

```
test_qa.head()
```

	textID	text	sentiment	question
0	f87dea47db	Last session of the day http://twitpic.com/67ezh	neutral	Why is this sentiment neutral?
1	96d74cb729	Shanghai is also really exciting (precisely -...	positive	Why is this sentiment positive?
2	eee518ae67	Recession hit Veronique Branquinho, she has to...	negative	Why is this sentiment negative?
3	01082688c6	happy bday!	positive	Why is this sentiment positive?
4	33987a8ee5	http://twitpic.com/4w75p - I like it!!	positive	Why is this sentiment positive?

## ▼ Implement Transformer question answering

References: [https://huggingface.co/docs/transformers/tasks/question\\_answering](https://huggingface.co/docs/transformers/tasks/question_answering)

<https://medium.com/mlearning-ai/question-answering-in-association-with-roberta-a11518e70507>

```
from transformers import AutoTokenizer
```

```
tokenizer_qa = AutoTokenizer.from_pretrained("distilbert-base-uncased")
```

```

/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
warnings.warn(
loading configuration file config.json from cache at /root/.cache/huggingface/hub/models--distilbert-base-uncased/snapshots/6cdc0aad91f5ae2e6712e91bc7b65d1cf5c05411/vocab.txt
Model config DistilBertConfig {
  "_name_or_path": "distilbert-base-uncased",
  "activation": "gelu",
  "architectures": [
    "DistilBertForMaskedLM"
  ],
  "attention_dropout": 0.1,
  "dim": 768,
  "dropout": 0.1,
  "hidden_dim": 3072,
  "initializer_range": 0.02,
  "max_position_embeddings": 512,
  "model_type": "distilbert",
  "n_heads": 12,
  "n_layers": 6,
  "pad_token_id": 0,
  "qa_dropout": 0.1,
  "seq_classif_dropout": 0.2,
  "sinusoidal_pos_embs": false,
  "tie_weights_": true,
  "transformers_version": "4.35.2",
  "vocab_size": 30522
}

loading file vocab.txt from cache at /root/.cache/huggingface/hub/models--distilbert-base-uncased/snapshots/6cdc0aad91f5ae2e6712e91bc7b65d1cf5c05411/vocab.txt
loading file tokenizer.json from cache at /root/.cache/huggingface/hub/models--distilbert-base-uncased/snapshots/6cdc0aad91f5ae2e6712e91bc7b65d1cf5c05411/tokenizer.json
loading file added_tokens.json from cache at None
loading file special_tokens_map.json from cache at None

```

```

loading file tokenizer_config.json from cache at /root/.cache/huggingface/hub/models--distilbert-base-uncased/snapshots/6cdc0aad91f5ae2e6712e91bc7b65d1c1
loading configuration file config.json from cache at /root/.cache/huggingface/hub/models--distilbert-base-uncased/snapshots/6cdc0aad91f5ae2e6712e91bc7b65d1c1
Model config DistilBertConfig {
  "_name_or_path": "distilbert-base-uncased",
  "activation": "gelu",
  "architectures": [
    "DistilBertForMaskedLM"
  ],
  "attention_dropout": 0.1,
  "dim": 768,
  "dropout": 0.1,
  "hidden_dim": 3072,
  "initializer_range": 0.02,
  "max_position_embeddings": 512,
  "model_type": "distilbert",
  "n_heads": 12,
  "n_layers": 6,
  "pad_token_id": 0,
  "qa_dropout": 0.1,
  "seq_classif_dropout": 0.2,
  "sinusoidal_pos_embs": false,
  "tie_weights": true
}

```

```

def preprocess_function(examples):
    questions = [q.strip() for q in examples["question"]]
    inputs = tokenizer_qa(
        questions,
        examples["text"],
        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]
        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
        if start_char >= 0: # Add this condition for cases that have empty text & selected_text
            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 or start_char < 0:
            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

```

```

pip install datasets

```

```

Requirement already satisfied: datasets in /usr/local/lib/python3.10/dist-packages (2.16.1)

```

```

Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from datasets) (3.13.1)

```

Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from datasets) (1.23.5)  
Requirement already satisfied: pyarrow>=8.0.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (10.0.1)  
Requirement already satisfied: pyarrow-hotfix in /usr/local/lib/python3.10/dist-packages (from datasets) (0.6)  
Requirement already satisfied: dill<0.3.8,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.3.7)  
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from datasets) (1.5.3)  
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (2.31.0)  
Requirement already satisfied: tqdm>=4.62.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (4.66.1)  
Requirement already satisfied: xxhash in /usr/local/lib/python3.10/dist-packages (from datasets) (3.4.1)  
Requirement already satisfied: multiprocessing in /usr/local/lib/python3.10/dist-packages (from datasets) (0.70.15)  
Requirement already satisfied: fsspec[http]<=2023.10.0,>=2023.1.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (2023.6.0)  
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from datasets) (3.9.3)  
Requirement already satisfied: huggingface-hub>=0.19.4 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.20.3)  
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from datasets) (23.2)  
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (6.0.1)  
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3.1)  
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (23.2.0)  
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.4.1)  
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6.0.4)  
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.9.4)  
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (4.0.3)  
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.4->datasets) (4.5.0)  
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (3.3.2)  
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (3.6)  
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (2.0.7)  
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (2023.11.17)  
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2.8.2)  
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2023.4)  
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas->datasets) (1.16.0)

```
def convert_answers(r):
    start = r[0]
    text = r[1]
    return {
        'answer_start': [start],
        'text': [text]
    }

from datasets import Dataset

skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=777)
for fold, (train_idx, val_idx) in enumerate(skf.split(train_qa, train_qa['sentiment'])):

    print('#'*25)
    print('### FOLD %i'%(fold+1))
    print('#'*25)

    if (fold == 0):
        continue;
    elif (fold == 2):
        break;

    train_df = train_qa.iloc[train_idx].copy()
    validation_df = train_qa.iloc[val_idx].copy()

    # train = train.sample(frac=1, random_state=42)
    train_df['answers'] = train_df[['answer_start', 'selected_text']].apply(convert_answers, axis=1)
    validation_df['answers'] = validation_df[['answer_start', 'selected_text']].apply(convert_answers, axis=1)

    train_dataset = Dataset.from_pandas(train_df)
    valid_dataset = Dataset.from_pandas(validation_df)

    tokenized_train_ds = train_dataset.map(preprocess_function, batched=True, remove_columns=train_dataset.column_names)
    tokenized_valid_ds = valid_dataset.map(preprocess_function, batched=True, remove_columns=valid_dataset.column_names)

    #####
    ### FOLD 1
    #####
    ### FOLD 2
    #####

    Map: 100%                               21985/21985 [00:11<00:00, 1663.06 examples/s]

    Map: 100%                               5496/5496 [00:03<00:00, 1463.23 examples/s]

    #####
    ### FOLD 3
    #####

train_dataset[0]
```

```
{'textID': 'cb774db0d1',
 'text': 'I'd have responded, if I were going',
 'selected_text': 'I'd have responded, if I were going',
 'sentiment': 'neutral',
 'question': 'Why is this sentiment neutral?',
 'answer_start': 0,
 'answers': {'answer_start': [0],
 'text': ['I'd have responded, if I were going']}},
 '_index_level_0_': 0}
```

```
from transformers import DefaultDataCollator
```

```
data_collator = DefaultDataCollator()
```

```
from transformers import AutoModelForQuestionAnswering, TrainingArguments, Trainer
```

```
access_token = "hf_xbArKpX0Eb0cUiqHvqMLeTolpwJBFtzgkv"
```

```
model = AutoModelForQuestionAnswering.from_pretrained("distilbert-base-uncased", token=access_token)
```

```
loading configuration file config.json from cache at /root/.cache/huggingface/hub/models--distilbert-base-uncased/snapshots/6cdc0aad91f5ae2e6712e91bc7b65d1c
Model config DistilBertConfig {
  "_name_or_path": "distilbert-base-uncased",
  "activation": "gelu",
  "architectures": [
    "DistilBertForMaskedLM"
  ],
  "attention_dropout": 0.1,
  "dim": 768,
  "dropout": 0.1,
  "hidden_dim": 3072,
  "initializer_range": 0.02,
  "max_position_embeddings": 512,
  "model_type": "distilbert",
  "n_heads": 12,
  "n_layers": 6,
  "pad_token_id": 0,
  "qa_dropout": 0.1,
  "seq_classif_dropout": 0.2,
  "sinusoidal_pos_embs": false,
  "tie_weights_": true,
  "transformers_version": "4.35.2",
  "vocab_size": 30522
}
```

```
loading weights file model.safetensors from cache at /root/.cache/huggingface/hub/models--distilbert-base-uncased/snapshots/6cdc0aad91f5ae2e6712e91bc7b65d1c
Some weights of the model checkpoint at distilbert-base-uncased were not used when initializing DistilBertForQuestionAnswering: ['vocab_layer_norm.weight',
- This IS expected if you are initializing DistilBertForQuestionAnswering from the checkpoint of a model trained on another task or with another architecture
- This IS NOT expected if you are initializing DistilBertForQuestionAnswering from the checkpoint of a model that you expect to be exactly identical (initialia
Some weights of DistilBertForQuestionAnswering were not initialized from the model checkpoint at distilbert-base-uncased and are newly initialized: ['qa_out
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```
pip install transformers[torch]
```

```
Requirement already satisfied: transformers[torch] in /usr/local/lib/python3.10/dist-packages (4.35.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (3.13.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.16.4 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (0.20.3)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (1.23.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (23.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (6.0.1)
Requirement already satisfied: regex<=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (2019.12.17)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (2.31.0)
Requirement already satisfied: tokenizers<0.19,>=0.14 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (0.15.1)
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (0.4.2)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (4.66.1)
Requirement already satisfied: torch!=1.12.0,>=1.10 in /usr/local/lib/python3.10/dist-packages (from transformers[torch]) (2.1.0+cu121)
Collecting accelerate>=0.20.3 (from transformers[torch])
  Downloading accelerate-0.26.1-py3-none-any.whl (270 kB)
270.9/270.9 kB 10.6 MB/s eta 0:00:00
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from accelerate>=0.20.3->transformers[torch]) (5.9.5)
Requirement already satisfied: fsspec<=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.16.4->transformers[torch]) (2023.6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.16.4->transformers[torch]) (4.6.1)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch!=1.12.0,>=1.10->transformers[torch]) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch!=1.12.0,>=1.10->transformers[torch]) (3.2.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch!=1.12.0,>=1.10->transformers[torch]) (3.1.3)
Requirement already satisfied: triton==2.1.0 in /usr/local/lib/python3.10/dist-packages (from torch!=1.12.0,>=1.10->transformers[torch]) (2.1.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->transformers[torch]) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers[torch]) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers[torch]) (2.0.7)
Requirement already satisfied: certifi<=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers[torch]) (2023.11.17)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch!=1.12.0,>=1.10->transformers[torch]) (2.1.4)
```

Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch!=1.12.0,>=1.10->transformers[torch]) (1.3.0)  
Installing collected packages: accelerate  
Successfully installed accelerate-0.26.1

```
import accelerate
```

```
accelerate.__version__
```

```
'0.26.1'
```

```
# !pip install huggingface_hub
```

```
from huggingface_hub.hf_api import HfFolder; HfFolder.save_token('hf_xbArKpX0Eb0cUiqHvqMLeTolpwJBFtzgkv')
```

```
training_args = TrainingArguments(  
    output_dir=google_drive_path+"fold2/",  
    evaluation_strategy="epoch",  
    learning_rate=2e-5,  
    per_device_train_batch_size=16,  
    per_device_eval_batch_size=16,  
    num_train_epochs=3,  
    save_strategy="epoch",  
    weight_decay=0.01
```

```
)
```

```
trainer = Trainer(  
    model=model,  
    args=training_args,  
    train_dataset=tokenized_train_ds,  
    eval_dataset=tokenized_valid_ds,  
    tokenizer=tokenizer_qa,  
    data_collator=data_collator
```

```
)
```

PyTorch: setting up devices

The default value for the training argument `--report_to` will change in v5 (from all installed integrations to none). In v5, you will need to use `--report`



```
tf.experimental.numpy.experimental_enable_numpy_behavior()
```

```
trainer.train()
```

```

**** Running training ****
Num examples = 21,985
Num Epochs = 3
Instantaneous batch size per device = 16
Total train batch size (w. parallel, distributed & accumulation) = 16
Gradient Accumulation steps = 1
Total optimization steps = 4,125
Number of trainable parameters = 66,364,418

```

[4125/4125 47:19, Epoch 3/3]

Epoch	Training Loss	Validation Loss
1	1.368300	1.261462
2	1.174500	1.200260
3	1.045700	1.209938

```

**** Running Evaluation ****

```

```

Num examples = 5496
Batch size = 16
Saving model checkpoint to /content/drive/MyDrive/XAI/fold2/checkpoint-1375
Configuration saved in /content/drive/MyDrive/XAI/fold2/checkpoint-1375/config.json
Model weights saved in /content/drive/MyDrive/XAI/fold2/checkpoint-1375/pytorch_model.bin
tokenizer config file saved in /content/drive/MyDrive/XAI/fold2/checkpoint-1375/tokenizer_config.json
Special tokens file saved in /content/drive/MyDrive/XAI/fold2/checkpoint-1375/special_tokens_map.json

```

```

**** Running Evaluation ****

```

```

Num examples = 5496
Batch size = 16
Saving model checkpoint to /content/drive/MyDrive/XAI/fold2/checkpoint-2750
Configuration saved in /content/drive/MyDrive/XAI/fold2/checkpoint-2750/config.json
Model weights saved in /content/drive/MyDrive/XAI/fold2/checkpoint-2750/pytorch_model.bin
tokenizer config file saved in /content/drive/MyDrive/XAI/fold2/checkpoint-2750/tokenizer_config.json
Special tokens file saved in /content/drive/MyDrive/XAI/fold2/checkpoint-2750/special_tokens_map.json

```

```

**** Running Evaluation ****

```

```

Num examples = 5496
Batch size = 16
Saving model checkpoint to /content/drive/MyDrive/XAI/fold2/checkpoint-4125
Configuration saved in /content/drive/MyDrive/XAI/fold2/checkpoint-4125/config.json
Model weights saved in /content/drive/MyDrive/XAI/fold2/checkpoint-4125/pytorch_model.bin
tokenizer config file saved in /content/drive/MyDrive/XAI/fold2/checkpoint-4125/tokenizer_config.json
Special tokens file saved in /content/drive/MyDrive/XAI/fold2/checkpoint-4125/special_tokens_map.json

```

Training completed. Do not forget to share your model on [huggingface.co/models](https://huggingface.co/models) =)

```

TrainOutput(global_step=4125, training_loss=1.2455519168738163, metrics={'train_runtime': 2841.7764,
'train_samples_per_second': 23.209, 'train_steps_per_second': 1.452, 'total_flos': 6462918909457920.0,
'train_loss': 1.2455519168738163, 'epoch': 3.0})

```

```

# trainer.save_model(google_drive_path+"fold2/my-fold2-model/")

```

## ▼ Metric

```

def jaccard(str1, str2):
    a = set(str1.lower().split())
    b = set(str2.lower().split())
    if (len(a)==0) & (len(b)==0): return 0.5
    c = a.intersection(b)
    return float(len(c)) / (len(a) + len(b) - len(c))

```

## ▼ Evaluate model with Jaccard

```

from transformers import AutoTokenizer
import torch
from transformers import AutoModelForQuestionAnswering

tokenizer = AutoTokenizer.from_pretrained(google_drive_path+"fold2/checkpoint-4125")
fold2_model = AutoModelForQuestionAnswering.from_pretrained(google_drive_path+"fold2/checkpoint-4125")

```

```

all = []
all_st = []
jac = []
for example in valid_dataset:
    question = example['question']
    context = example['text']
    inputs = tokenizer(question, context, return_tensors="pt")

    with torch.no_grad():
        outputs = fold2_model(**inputs)

    answer_start_index = outputs.start_logits.argmax()
    answer_end_index = outputs.end_logits.argmax()

    if answer_start_index > answer_end_index:
        st = example['text'] # IMPROVE CV/LB with better choice here
    else:
        # text1 = " "+" ".join(context.split())
        # enc = tokenizer.encode(text1)
        # st = tokenizer.decode(enc.ids[a-1:b])
        predict_answer_tokens = inputs.input_ids[0, answer_start_index : answer_end_index + 1]
        st = tokenizer.decode(predict_answer_tokens)
        st = st.replace(' [SEP]', '') # This is for some selected text have [SEP] at the beginning
    all_st.append(st)
    all.append(jaccard(st, example['selected_text']))
jac.append(np.mean(all))

```

```

loading file vocab.txt
loading file tokenizer.json
loading file added_tokens.json
loading file special_tokens_map.json
loading file tokenizer_config.json
loading configuration file /content/drive/MyDrive/XAI/fold2/checkpoint-4125/config.json
Model config DistilBertConfig {
  "_name_or_path": "/content/drive/MyDrive/XAI/fold2/checkpoint-4125",
  "activation": "gelu",
  "architectures": [
    "DistilBertForQuestionAnswering"
  ],
  "attention_dropout": 0.1,
  "dim": 768,
  "dropout": 0.1,
  "hidden_dim": 3072,
  "initializer_range": 0.02,
  "max_position_embeddings": 512,
  "model_type": "distilbert",
  "n_heads": 12,
  "n_layers": 6,
  "pad_token_id": 0,
  "qa_dropout": 0.1,
  "seq_classif_dropout": 0.2,
  "sinusoidal_pos_embs": false,
  "tie_weights": true,
  "torch_dtype": "float32",
  "transformers_version": "4.35.2",
  "vocab_size": 30522
}

```

```

loading weights file /content/drive/MyDrive/XAI/fold2/checkpoint-4125/model.safetensors
All model checkpoint weights were used when initializing DistilBertForQuestionAnswering.

```

All the weights of DistilBertForQuestionAnswering were initialized from the model checkpoint at /content/drive/MyDrive/XAI/fold2/checkpoint-4125. If your task is similar to the task the model of the checkpoint was trained on, you can already use DistilBertForQuestionAnswering for predictions without f

```
print(jac)
```

```
[0.5718368328426511]
```

```
# Save model extracted text to the df
```

```
validation_df['model_selected_text'] = all_st
```



## ✓ Apply SHAP QA on validation dataset

[https://shap.readthedocs.io/en/latest/example\\_notebooks/text\\_examples/question\\_answering/Explaining%20a%20Question%20Answering%20Transformers%20Model.html](https://shap.readthedocs.io/en/latest/example_notebooks/text_examples/question_answering/Explaining%20a%20Question%20Answering%20Transformers%20Model.html)

```
pip install shap
```

```
Collecting shap
  Downloading shap-0.44.1-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (535 kB)
    535.7/535.7 kB 13.4 MB/s eta 0:00:00
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from shap) (1.23.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from shap) (1.11.4)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (from shap) (1.2.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from shap) (1.5.3)
Requirement already satisfied: tqdm>=4.27.0 in /usr/local/lib/python3.10/dist-packages (from shap) (4.66.1)
Requirement already satisfied: packaging>20.9 in /usr/local/lib/python3.10/dist-packages (from shap) (23.2)
Collecting slicer==0.0.7 (from shap)
  Downloading slicer-0.0.7-py3-none-any.whl (14 kB)
Requirement already satisfied: numba in /usr/local/lib/python3.10/dist-packages (from shap) (0.58.1)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.10/dist-packages (from shap) (2.2.1)
Requirement already satisfied: llvmlite<0.42,>=0.41.0dev0 in /usr/local/lib/python3.10/dist-packages (from numba->shap) (0.41.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas->shap) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->shap) (2023.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->shap) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->shap) (3.2.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas->shap) (1.16.0)
Installing collected packages: slicer, shap
Successfully installed shap-0.44.1 slicer-0.0.7
```

```
question_answerer = pipeline("question-answering", model=google_drive_path+'fold2/checkpoint-4125/')
```

```
loading configuration file /content/drive/MyDrive/XAI/fold2/checkpoint-4125/config.json
```

```
Model config DistilBertConfig {
  "_name_or_path": "/content/drive/MyDrive/XAI/fold2/checkpoint-4125/",
  "activation": "gelu",
  "architectures": [
    "DistilBertForQuestionAnswering"
  ],
  "attention_dropout": 0.1,
  "dim": 768,
  "dropout": 0.1,
  "hidden_dim": 3072,
  "initializer_range": 0.02,
  "max_position_embeddings": 512,
  "model_type": "distilbert",
  "n_heads": 12,
  "n_layers": 6,
  "pad_token_id": 0,
  "qa_dropout": 0.1,
  "seq_classif_dropout": 0.2,
  "sinusoidal_pos_embs": false,
  "tie_weights": true,
  "torch_dtype": "float32",
  "transformers_version": "4.35.2",
  "vocab_size": 30522
}
```

```
loading configuration file /content/drive/MyDrive/XAI/fold2/checkpoint-4125/config.json
```

```
Model config DistilBertConfig {
  "_name_or_path": "/content/drive/MyDrive/XAI/fold2/checkpoint-4125/",
  "activation": "gelu",
  "architectures": [
    "DistilBertForQuestionAnswering"
  ],
  "attention_dropout": 0.1,
  "dim": 768,
  "dropout": 0.1,
  "hidden_dim": 3072,
  "initializer_range": 0.02,
  "max_position_embeddings": 512,
  "model_type": "distilbert",
  "n_heads": 12,
  "n_layers": 6,
  "pad_token_id": 0,
  "qa_dropout": 0.1,
  "seq_classif_dropout": 0.2,
  "sinusoidal_pos_embs": false,
  "tie_weights": true,
  "torch_dtype": "float32",
  "transformers_version": "4.35.2",
  "vocab_size": 30522
}
```

loading weights file /content/drive/MyDrive/XAI/fold2/checkpoint-4125/model.safetensors  
All model checkpoint weights were used when initializing DistilBertForQuestionAnswering.

All the weights of DistilBertForQuestionAnswering were initialized from the model checkpoint at /content/drive/MyDrive/XAI/fold2/checkpoint-4125/.

```
def make_answer_scorer(answers):
    def f(questions):
        out = []
        for q in questions:
            question, context = q.split("[SEP]")
            results = question_answerer(question, context, topk=20)
            values = []
            for answer in answers:
                value = 0
                for result in results:
                    if result["answer"] == answer:
                        value = result["score"]
                        break
                values.append(value)
            out.append(values)
        return out

    f.output_names = answers
    return f
```

validation\_df.loc[:,['text','selected\_text', 'sentiment','model\_selected\_text']]

	text	selected_text	sentiment	model_selected_text
1	Sooo SAD I will miss you here in San Diego!!!	Sooo SAD	negative	##o sad
2	my boss is bullying me...	bullying me	negative	bullying
6	2am feedings for the baby are fun when he is a...	fun	positive	are fun
12	My Sharpie is running DANGERously low on ink	DANGERously	negative	dangerously low on ink
14	test test from the LG enV2	test test from the LG enV2	neutral	test test from the lg env2
...	...	...	...	...
27462	Just back from bingo w/family -- I won over \$1...	Fun	positive	won over \$ 1100! fun night
27463	LIKE DREW SAID 'GIVE TC A CHANCE' WE WILL MIS...	MISS	negative	will miss
27467	morning twit-friends! welcome to my new followers	welcome	positive	! welcome

```
import shap
# Negative text
new_string = validation_df.loc[46,'question'] + "[SEP]" + validation_df.loc[46,'text']

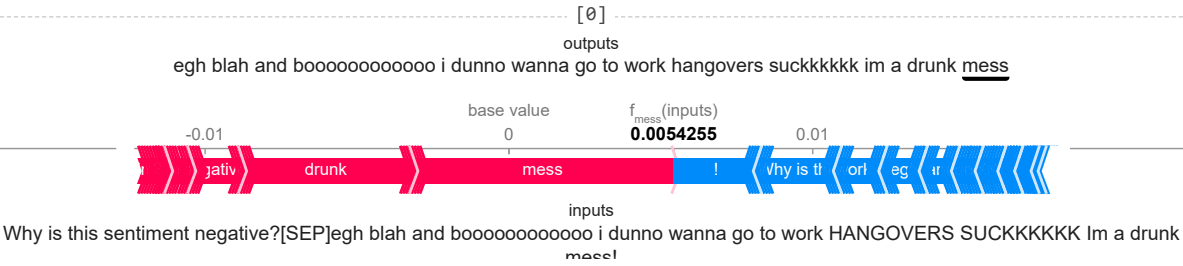
our_train_data = []
our_train_data.append(new_string)

f_answers = make_answer_scorer(validation_df.loc[46,'model_selected_text'].split())
explainer_answers = shap.Explainer(f_answers, tokenizer)
shap_values_answers = explainer_answers(our_train_data)

print(validation_df.loc[46,'selected_text'])
shap.plots.text(shap_values_answers)
```

topk parameter is deprecated, use top\_k instead  
Disabling tokenizer parallelism, we're using DataLoader multithreading already  
topk parameter is deprecated, use top\_k instead  
topk parameter is deprecated, use top\_k instead

PartitionExplainer explainer: 2it [00:44, 44.78s/it]  
SUCKKKKKK



```
# Positive text
new_string = validation_df.loc[6,'question'] + "[SEP]" + validation_df.loc[6,'text']

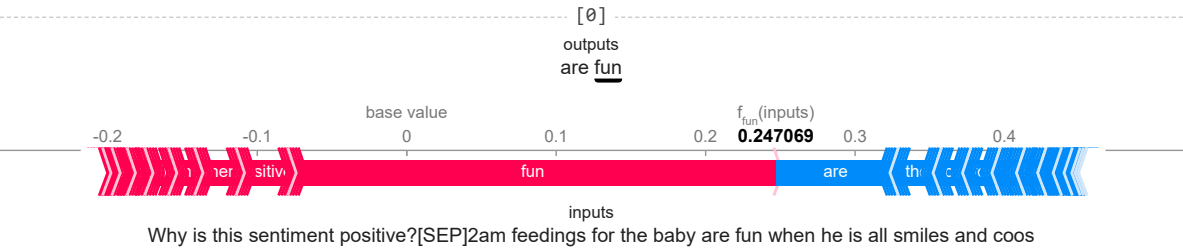
our_train_data = []
our_train_data.append(new_string)

f_answers = make_answer_scorer(validation_df.loc[6,'model_selected_text'].split())
explainer_answers = shap.Explainer(f_answers, tokenizer)
shap_values_answers = explainer_answers(our_train_data)

print(validation_df.loc[6,'selected_text'])
shap.plots.text(shap_values_answers)
```

topk parameter is deprecated, use top\_k instead

PartitionExplainer explainer: 2it [00:48, 48.87s/it] fun



▼ Prepare dataset for testing

```
def prepare_validation_features(examples):
```

```
    questions = [q.strip() for q in examples["question"]]

test_dataset = Dataset.from_pandas(test_qa)
test_features = test_dataset.map(
    prepare_validation_features,
    batched=True,
    remove_columns=test_dataset.column_names
)

test_feats_small = test_features.map(lambda example: example, remove_columns=['example_id', 'offset_mapping'])
```

```
Map: 100% 3534/3534 [00:01<00:00, 1622.08 examples/s]

Map: 100% 3534/3534 [00:00<00:00, 18438.30 examples/s]

for i in range(len(inputs["input_ids"])):
```

▼ Show extracted text for test dataset

```
all_st_test = []
for example in test_dataset:
    question = example['question']
    context = example['text']
    inputs = tokenizer(question, context, return_tensors="pt")

    with torch.no_grad():
        outputs = fold2_model(**inputs)

    answer_start_index = outputs.start_logits.argmax()
    answer_end_index = outputs.end_logits.argmax()

    if answer_start_index>answer_end_index:
        st = example['text'] # IMPROVE CV/LB with better choice here
    else:
        predict_answer_tokens = inputs.input_ids[0, answer_start_index : answer_end_index + 1]
        st = tokenizer.decode(predict_answer_tokens)
        st = st.replace("[SEP]", "")
    all_st_test.append(st)

test['selected_text'] = all_st_test
test[['textID','selected_text']].to_csv('submission.csv',index=False)
pd.set_option('max_colwidth', 60)
test.sample(25)
```

	textID	text	sentiment	selected_text
1885	839095ea38	http://twitpic.com/66nbd - Ready for our 3D Jonas Brothe...	neutral	ready for our 3d jonas brothers experience. real brother...
3360	a29a6f5c41	I forgot, Happy Mom`s day.	positive	happy mom`s day.
222	37ffa83550	thank you!! ooh I see you`ve read Desert Islands http:/...	positive	###k great book
2384	05198b8107	@ my sisters crying my eyes out, hubby called from Iraq,...	negative	crying my eyes out,
318	0ce30035ac	I just walked into work, all the while thinking that I w...	neutral	i just walked into work, all the while thinking that i w...
3276	d36296a726	MAKE ME ONE! I`m still craving shrimp	neutral	make me one! i`m still craving shrimp