Submission for
Exercise task 8
of UTU course TKO_8964-3006
Textual Data Analysis
by Botond Ortutay

Instructions:

In this exercise you will test a small idea:

"Let us assume we train a model which receives a question and a text segment on its input, and predicts YES/NO whether the text segment contains the answer to the question. It should then be so that if the answer is YES, the explanation of the prediction should point out to the answer in the text."

I tested this and, well, seems like this small idea works and now your job is to replicate it, i.e. arrive at this output:

I trained a BERT model for you for that task, and you can find it here: http://dl.turkunlp.org/TKO_8964_2023/english-binarized-weighted.model.tgz

It is taking its input in the form "[CLS] question [SEP] context [SEP]" and the output has two logit values, the first one is for the negative class (question not answered) and the second one for the positive class (question answered). You can download the model, unpack it, and run as follows:

```
MODEL_NAME = 'english-binarized-weighted.model'
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
model = AutoModelForSequenceClassification.from_pretrained(MODEL_NAME)
tokenized = tokenizer(text=question, text_pair=context, return_tensors='pt')
prediction = model(**tokenized)
```

The rest you should be able to base off the example notebook we had on the lecture, it is basically the exact same code and you should be able to replicate the result for at least the Q-A pair in the screenshot above.

Solution:

Library & environment setup:

In [1]:
```
from captum.attr import LayerIntegratedGradients
from IPython.display import display, Markdown
import numpy as np
import pandas as pd
import random
import torch
from transformers import AutoTokenizer, AutoModelForSequenceClassification

# checking GPU availability
print(torch.cuda.is_available()) # should output True
```

```
True
```

Model downloading & unpacking:

In [2]:
```
#NOTE: bash code here
# Downloading the model
!wget http://dl.turkunlp.org/TKO_8964_2023/english-binarized-weighted.model.tgz
# Locally extracting .tar.gz package with a single command:
!tar -xzvf english-binarized-weighted.model.tgz
# Printing local datastructure to make sure it is now on disk
!echo
!tree english-binarized-weighted.model
```

```
7[Files: 0  Bytes: 0  [0 B/s] Re]87[http://dl.turkunlp.org/TKO_896]87Saving 'english-binarized-weighted.model.tg
z'
87english-binarized-we   1% [>                          ]    6.11M    --.-KB/s87[Files: 0  Bytes: 0  [0 B/s]
Re]87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we   3% [>                           ]   12.91M    6.
79MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we   5% [>                              ]   19.49M
6.68MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we   6% [=>                             ]   26.60M
6.82MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we   8% [=>                             ]   33.86M
6.93MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  10% [==>                            ]   40.58M
6.89MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  12% [==>                            ]   47.01M
6.81MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  13% [===>                           ]   52.93M
6.68MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  15% [===>                           ]   58.28M
```

```
6.51MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  16% [====>                          ]   64.41M
6.47MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  18% [====>                          ]   70.39M
6.42MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  20% [=====>                         ]   76.80M
6.42MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  21% [=====>                         ]   83.06M
6.41MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  23% [======>                        ]   89.83M
6.43MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  25% [======>                        ]   96.25M
6.43MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  26% [=======>                       ]  102.41M
6.41MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  28% [=======>                       ]  109.71M
6.47MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  30% [========>                      ]  116.53M
6.49MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  31% [========>                      ]  122.52M
6.46MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  33% [=========>                     ]  129.82M
6.50MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  35% [=========>                     ]  134.72M
6.42MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  36% [==========>                    ]  141.39M
6.44MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  38% [==========>                    ]  149.16M
6.50MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  40% [===========>                   ]  155.94M
6.51MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  42% [===========>                   ]  162.43M
6.50MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  44% [============>                  ]  169.48M
6.52MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  45% [============>                  ]  175.45M
6.47MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  47% [=============>                 ]  181.86M
6.43MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  48% [=============>                 ]  186.75M
6.35MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  50% [==============>                ]  193.83M
6.38MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  52% [==============>                ]  200.26M
6.40MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  54% [===============>               ]  208.76M
6.54MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  56% [===============>               ]  215.17M
6.55MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  57% [================>              ]  221.62M
6.57MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  59% [================>              ]  227.16M
6.53MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  60% [=================>             ]  232.65M
6.50MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  62% [=================>             ]  237.98M
6.44MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  64% [==================>            ]  245.45M
6.48MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  65% [==================>            ]  251.99M
6.50MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  67% [===================>           ]  258.27M
6.45MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  68% [===================>           ]  262.78M
6.35MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  69% [====================>          ]  268.36M
6.33MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  71% [====================>          ]  275.39M
6.32MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  73% [=====================>         ]  282.26M
6.41MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  75% [=====================>         ]  288.74M
6.40MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  76% [======================>        ]  294.93M
6.33MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  78% [======================>        ]  301.97M
6.34MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  80% [=======================>       ]  307.88M
6.32MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  81% [=======================>       ]  313.39M
6.25MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  83% [========================>      ]  319.07M
6.24MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  85% [========================>      ]  327.04M
6.31MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  86% [=========================>     ]  332.19M
6.32MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  87% [=========================>     ]  337.21M
6.23MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  89% [==========================>    ]  344.53M
6.27MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  91% [==========================>    ]  350.83M
6.17MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  93% [===========================>   ]  357.57M
6.19MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  94% [===========================>   ]  363.66M
6.17MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  96% [============================>  ]  368.97M
6.16MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  97% [============================> ]  375.07M
6.19MB/s87[Files: 0  Bytes: 0  [0 B/s] Re]87english-binarized-we  99% [============================> ]  382.15M
6.26MB/s87english-binarized-we 100% [=============================>]  383.50M    6.21MB/s87HTTP response 200 OK
[http://dl.turkunlp.org/TKO_8964_2023/english-binarized-weighted.model.tgz]
87english-binarized-we 100% [=============================>]  383.50M    6.21MB/s87[Files: 1  Bytes: 383.50M [6.
36]8english-binarized-weighted.model/
english-binarized-weighted.model/training_args.bin
english-binarized-weighted.model/pytorch_model.bin
english-binarized-weighted.model/tokenizer.json
english-binarized-weighted.model/vocab.txt
english-binarized-weighted.model/config.json
english-binarized-weighted.model/special_tokens_map.json
english-binarized-weighted.model/tokenizer_config.json

english-binarized-weighted.model
├── config.json
├── pytorch_model.bin
├── special_tokens_map.json
├── tokenizer_config.json
├── tokenizer.json
├── training_args.bin
└── vocab.txt

1 directory, 7 files
```

Functions for later use:

In [3]:
```python
# Forwarding function (needed for LayerIntegratedGradients)
def forwarder(inputs, token_type_ids, attention_mask):
    pred = model(inputs, token_type_ids=token_type_ids, attention_mask=attention_mask)
    return pred.logits #return the output of the classification layer
```

Importing model to Python:

In [4]:
```
MODEL_NAME = 'english-binarized-weighted.model'
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
model = AutoModelForSequenceClassification.from_pretrained(MODEL_NAME)
```

```
2025-02-11 22:03:23.265762: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:477] Unable to register cu
FFT factory: Attempting to register factory for plugin cuFFT when one has already been registered
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1739304203.282491   59187 cuda_dnn.cc:8310] Unable to register cuDNN factory: Attempting to register
factory for plugin cuDNN when one has already been registered
E0000 00:00:1739304203.287648   59187 cuda_blas.cc:1418] Unable to register cuBLAS factory: Attempting to regist
er factory for plugin cuBLAS when one has already been registered
2025-02-11 22:03:23.305199: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optim
ized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate com
piler flags.
```

Function for analyzing question-context pairs

Objectices:

1. Determine whether context has answer to question or not
2. If context has answer to question: where within the context is the answer?

In [5]:
```
"""
Function for analyzing question-context pairs

Objectices:
 1. Determine whether context has answer to question or not
 2. If context has answer to question: where within the context is the answer?
---
In:
question     str                             question in question-context pair
context      str                             long string where we search for the answer
model        BertForSequenceClassification   bert model we use to perform analysis
tokenizer    BertTokenizerFast               tokenizer to use with model
"""
def qAndCAnalyzer(question,context,model,tokenizer):
    tokenized = tokenizer(text=question, text_pair=context, return_tensors='pt')
    prediction = model(**tokenized)

    # Does the context not answer the question? (from prediction output)
    if (prediction["logits"][0][0] > prediction["logits"][0][1]):
        dString = 'Context doesn\'t have an answer to your question!'

    # If the context has an answer, we move on to further analysis
    else:
        # Tokenized output to tensor tuple (currently it's <class 'transformers.tokenization_utils_base.BatchEn
        tokenizedTT = (tokenized["input_ids"],tokenized["token_type_ids"],tokenized["attention_mask"])

        # New tensor tuple with padding characters, equal in length to tokenizedTT
        ref = tokenizer(" ".join(["[PAD]"]*(len(tokenized["input_ids"][0])-2)),return_tensors="pt")    # One pa
        baselineTT = (ref["input_ids"],ref["token_type_ids"],ref["attention_mask"])

        # setting up gradient system we'll use to calculate per-token attributes. Uses forwarding function from
        gradients = LayerIntegratedGradients(forwarder, model.bert.embeddings)

        # Using our gradients to calculate attribute scores for our inputs
        attrs = gradients.attribute(inputs=tokenizedTT, baselines = baselineTT, return_convergence_delta=False,

        # Math magic to scale the numbers
        attrs = attrs.sum(dim=-1).squeeze(0)

        # Results to numpy format for easier handling below
        attrNp = attrs.numpy()

        # Generating fancy output. Text bg will be green if a token is relevant to the question (the darker the
        dString = ''
        k = 0
        for i in tokenized["input_ids"]:
            for j in tokenizer.convert_ids_to_tokens(i):
                if(attrNp[k] > np.percentile(attrNp, 99)):
                    dString += '<mark style="background-color:#7cb342">' + j + '</mark> '
                elif(attrNp[k] > np.percentile(attrNp, 95)):
                    dString += '<mark style="background-color:#aed581">' + j + '</mark> '
                elif(attrNp[k] > np.percentile(attrNp, 91)):
                    dString += '<mark style="background-color:#dcedc8">' + j + '</mark> '
                elif(attrNp[k] < np.percentile(attrNp, 1)):
                    dString += '<mark style="background-color:#ff0000">' + j + '</mark> '
                elif(attrNp[k] < np.percentile(attrNp, 4)):
                    dString += '<mark style="background-color:#ff4747">' + j + '</mark> '
                elif(attrNp[k] < np.percentile(attrNp, 9)):
                    dString += '<mark style="background-color:#ff8f8f">' + j + '</mark> '
```

```
                else:
                    dString += j + " "
                k += 1
    # Fancy output
    display(Markdown(dString))
```

Testing with example provided in instructions:

In [6]:
```
question = "When was University of Turku founded?"
context = "The University of Turku (Finnish: Turun yliopisto, in Swedish: Åbo universitet), located in Turku in

qAndCAnalyzer(question,context,model,tokenizer)
```

[CLS] When was University of Tu ##rk ##u founded ? [SEP] The University of Tu ##rk ##u (Finnish : Tu ##run y ##lio ##pis ##to , in Swedish : Å ##bo un ##ivers ##ite ##t) , located in Tu ##rk ##u in so ##ut ##wes ##tern Finland , is the third largest university in the country as measured by student enrollment , after the University of Helsinki and Tam ##per ##e University . It is a multi ##disciplinary university with eight faculties . It was established in 1920 and also has facilities at Ra ##uma , Po ##ri , Ke ##vo and Se ##ili . The university is a member of the Co ##im ##bra Group and the European Campus of City - Un ##iv ##iers ##ities (EC ##2 ##U) . [SEP]

Testing with context that doesn't have the answer:

In [7]:
```
question = "When was the declaration of Independence signed?"
context = "The Sucunduri Formation is a Neoproterozoic geologic formation in Brazil. While reports made in the

qAndCAnalyzer(question,context,model,tokenizer)
```

Context doesn't have an answer to your question!

Testing with random samples:

In [8]:
```
"""
Let's create a very small dataset of 20 questions-context pairs
10 of the contexts have the answers and 10 don't
"""
qcPairs = pd.DataFrame([
    ["When was University of Turku founded?","The University of Turku (Finnish: Turun yliopisto, in Swedish: Åb
    ["When was the declaration of Independence signed?","The Sucunduri Formation is a Neoproterozoic geologic f
    ["What kind of seasoning to use for Chicken Fajitas?","Sheet Pan Chicken Fajitas <br><br> 1. Preheat the ov
    ["Where does the US Route 66 start?","Illinois Route 84 (Route 84 or IL 84) is a long state highway that ru
    ["Which historical event did Leo Alexander play a critical role in?","Leo Alexander (October 11, 1905 – July
    ["What is the capital of Denmark?","Cogenhoe United Football Club is a football club based in Cogenhoe, nea
    ["Why did the French government reward some of its citizens after World War I?","During the war of 1914-191
    ["What is the biggest music publishing company in India?","Zubeen Garg is an Indian singer, music director,
    ["Who was Lolita Chammah?","Chammah is a surname. Notable people with the surname include: <br> - Itai Cham
    ["What is the capital of Poland?","The poiseuille (symbol Pl) has been proposed as a derived SI unit of dyn
    ["Which album did the TLC song 'Waterfalls' appear on?","'Waterfalls' is a song by American hip-hop group T
    ["What caused the 2016 Killer Clowns trend?","The Anzick site (24PA506), located adjacent to Flathead Creek
    ["What is a common nickname for Pimento cheese?","Sobchak: You been doing this long? I just assume we're al
    ["What's the most popular TV series in Pakistan?","Ghalati is a 2019 Pakistani romantic drama television se
    ["What kind of plant is Cypreus Niveus?","Cyperus niveus is a species of sedge that is native to parts of A
    ["Where in India is Rajasthan located?","Jaydeep Bihani is an Indian politician serving as a member of the
    ["Which TV network did the show 'Orleans' air on?","Orleans is an American drama television series created
    ["Who is the most famous football player in the Netherlands?","Raoul Henar (born 4 August 1972) is a Dutch
    ["Which street is the San Bernardino Valley campus located on?","KVCR-DT (channel 24) is a PBS member televi
    ["Who owns the Mexican TV station Las Estrellas?","M.D.: Life on the Line (Spanish: Médicos, línea de vida)
], columns=["question","context","answer in context?","answer"])
for i in random.sample(range(0, 19), 5):
    dString = "Selecting random question-context pair for analysis.<br>**Question:** " + qcPairs["question"].lo
    if(qcPairs["answer in context?"].loc[qcPairs.index[i]]):
        dString += "The answer to the question **is** in the context.<br> **Expected answer:** "  + qcPairs["an
    else:
        dString += "The answer to the question **is not** in the context.<br>"
    dString += "**Analysis results:**<br>"
    display(Markdown(dString))
    qAndCAnalyzer(qcPairs["question"].loc[qcPairs.index[i]],qcPairs["context"].loc[qcPairs.index[i]],model,toke
```

Selecting random question-context pair for analysis.
Question: What caused the 2016 Killer Clowns trend?
The answer to the question **is not** in the context.
Analysis results:

Context doesn't have an answer to your question!

Selecting random question-context pair for analysis.
Question: Where does the US Route 66 start?
The answer to the question **is not** in the context.
Analysis results:

Context doesn't have an answer to your question!

Selecting random question-context pair for analysis.
Question: When was University of Turku founded?
The answer to the question **is** in the context.
Expected answer: 1920
Analysis results:

[CLS] When was University of Tu ##rk ##u founded ? [SEP] The University of Tu ##rk ##u (Finnish : Tu ##run y ##lio ##pis ##to , in Swedish : Å ##bo un ##ivers ##ite ##t) , located in Tu ##rk ##u in so ##ut ##wes ##tern Finland , is the third largest university in the country as measured by student enrollment , after the University of Helsinki and Tam ##per ##e University . It is a multi ##disciplinary university with eight faculties . It was established in 1920 and also has facilities at Ra ##uma , Po ##ri , Ke ##vo and Se ##ili . The university is a member of the Co ##im ##bra Group and the European Campus of City - Un ##iv ##iers ##ities (EC ##2 ##U) . [SEP]

Selecting random question-context pair for analysis.
Question: What is the biggest music publishing company in India?
The answer to the question **is not** in the context.
Analysis results:

Context doesn't have an answer to your question!

Selecting random question-context pair for analysis.
Question: Where in India is Rajasthan located?
The answer to the question **is not** in the context.
Analysis results:

Context doesn't have an answer to your question!