Historical Texts Summarizer/Modernizer

By: Landon Bragg & Ben Kim

Intro

The objective of this project is to develop a model which transforsm or summarizes historical texts into modern English. By achieving this, we make older texts more readable and comprehensible to a contemporary audience. This project explores the application of text summarization techniques as well as language transformation models. These utilize data derived from historical texts paired with professionally written summaries.

Client Needs

Goal:

We want to train a neural network to use weights and balances to comprehend old texts written in outdated language and then translate and summarize it to our client's needs.

Target Audience:

Our target audience would be scholars or students who need to understand historical texts, but might be struggling with a certain passage. Additionally, this would be a great way to teach students about the nuances of language by condensing complex and unique language into something more understandable.

Inspiration:

Our client is currently in the Great Texts course at Baylor and is tired of doing all of the reading and struggling to decipher old English. He wants our model to help him out with this problem in order to save him some time from the confusing reading.

Desired Output:

Summaries that maintain the key points and relevance of the original text. Summaries should be neutral, unbiased, and cover major facts. Summaries should be shorter than the original text.

Important Notes:

This model must ensure that it doesn't lack any main ideas from the original text. It still needs to summarize and shorten the original text, but it must not leave any major points out.

```
# This line must be run each time to run the code regardless of previous install
!pip install datasets
```

```
Collecting datasets
  Downloading datasets-3.1.0-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from datasets) (3.16.1)
Requirement already satisfied: numpy>=1.17 in
/usr/local/lib/python3.10/dist-packages (from datasets) (1.26.4)
Requirement already satisfied: pyarrow>=15.0.0 in
/usr/local/lib/python3.10/dist-packages (from datasets) (17.0.0)
Collecting dill<0.3.9,>=0.3.0 (from datasets)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
(from datasets) (2.2.2)
Requirement already satisfied: requests>=2.32.2 in
/usr/local/lib/python3.10/dist-packages (from datasets) (2.32.3)
Requirement already satisfied: tqdm>=4.66.3 in
/usr/local/lib/python3.10/dist-packages (from datasets) (4.66.6)
Collecting xxhash (from datasets)
  Downloading
  xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
  (12 kB)
Collecting multiprocess<0.70.17 (from datasets)
  Downloading multiprocess-0.70.16-py310-none-any.whl.metadata (7.2 kB)
Collecting fsspec<=2024.9.0,>=2023.1.0 (from
fsspec[http]<=2024.9.0,>=2023.1.0->datasets)
  Downloading fsspec-2024.9.0-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages
(from datasets) (3.10.10)
Requirement already satisfied: huggingface-hub>=0.23.0 in
/usr/local/lib/python3.10/dist-packages (from datasets) (0.24.7)
Requirement already satisfied: packaging in
/usr/local/lib/python3.10/dist-packages (from datasets) (24.1)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.10/dist-packages (from datasets) (6.0.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (2.4.3)
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) (24.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6.1.0)
```

```
Requirement already satisfied: yarl<2.0,>=1.12.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.17.1)
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>=0.23.0->datasets)
(4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets)
(2024.8.30)
Requirement already satisfied: python-dateutil>=2.8.2 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) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from
python-dateutil>=2.8.2->pandas->datasets) (1.16.0)
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from
yarl<2.0,>=1.12.0->aiohttp->datasets) (0.2.0)
Downloading datasets-3.1.0-py3-none-any.whl (480 kB)
                         480.6/480.6 kB 4.7 MB/s eta 0:00:00
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
                         116.3/116.3 kB 5.1 MB/s eta 0:00:00
Downloading fsspec-2024.9.0-py3-none-any.whl (179 kB)
                         179.3/179.3 kB 5.7 MB/s eta 0:00:00
Downloading multiprocess-0.70.16-py310-none-any.whl (134 kB)
                         134.8/134.8 kB 7.5 MB/s eta 0:00:00
Downloading
xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
                         194.1/194.1 kB 13.0 MB/s eta 0:00:00
Installing collected packages: xxhash, fsspec, dill, multiprocess, datasets
  Attempting uninstall: fsspec
    Found existing installation: fsspec 2024.10.0
    Uninstalling fsspec-2024.10.0:
      Successfully uninstalled fsspec-2024.10.0
ERROR: pip's dependency resolver does not currently take into account all the
packages that are installed. This behaviour is the source of the following
dependency conflicts.
```

```
gcsfs 2024.10.0 requires fsspec==2024.10.0, but you have fsspec 2024.9.0 which is
incompatible.
Successfully installed datasets-3.1.0 dill-0.3.8 fsspec-2024.9.0
multiprocess-0.70.16 xxhash-3.5.0
```

```python
import pandas as pd
from google.colab import drive

drive.mount('/content/drive')
dataset = pd.read_csv('/content/drive/My Drive/Colab
Notebooks/Data/HistoricalTextDataset.csv')

print(dataset.head())
dataset.info()

# Clean the dataset by stripping leading/trailing whitespace
dataset['train_data'] = dataset['train_data'].str.strip()
dataset['eval_data'] = dataset['eval_data'].str.strip()

print(dataset.head())
```

```
Mounted at /content/drive
                                          train_data  \
0  He said he the war-king would seek o'er the oc...
1  And thus lith Alison and Nicholas, In bisiness...
2  Therefore I intend to show a wonder on earth, ...
3  Thus have I described to you, as exactly as I ...
4  lovely Ladie rode him faire beside, Upon a low...

                                           eval_data
0  He said the king would search across the ocean...
1  And so Alison and Nicholas stayed together, en...
2  Therefore, I plan to reveal a wonder on earth,...
3  So I have described to you, as accurately as I...
4  A lovely lady rode beside him on a lowly donke...
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19 entries, 0 to 18
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   train_data  19 non-null     object
 1   eval_data   19 non-null     object
dtypes: object(2)
memory usage: 432.0+ bytes
                                          train_data  \
0  He said he the war-king would seek o'er the oc...
1  And thus lith Alison and Nicholas, In bisiness...
2  Therefore I intend to show a wonder on earth, ...
```

```
3  Thus have I described to you, as exactly as I ...
4  lovely Ladie rode him faire beside, Upon a low...

                                          eval_data
0  He said the king would search across the ocean...
1  And so Alison and Nicholas stayed together, en...
2  Therefore, I plan to reveal a wonder on earth,...
3  So I have described to you, as accurately as I...
4  A lovely lady rode beside him on a lowly donke...
```

Explanation of training data

The training data is a custom CSV file created from historical texts and their corresponding modern English summaries. Each row contains two columns:

train_data: A segment of historical text.

eval_data: The professionally summarized or modernized version of that text.

The dataset comprises 19 entries, each providing a unique instance of language transformation from archaic or complex English into a more accessible form. This small dataset serves as the foundation for model training and evaluation, aiming to capture the nuances of historical language while providing accurate and contextually appropriate summaries.

Instructions for running

- After running this block of code, click on the link that follows "wandb: You can find your API key in your browser here:"
- After clicking on the link create an account and copy and paste the API key into the text box which will allow the code to run.

```
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM, Trainer,
TrainingArguments
from datasets import Dataset

# Load pre-trained model and tokenizer
model_name = "t5-small"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)

def preprocess_function(examples):
    inputs = ["summarize: " + doc for doc in examples["train_data"]]
    model_inputs = tokenizer(inputs, max_length=512, truncation=True,
padding="max_length")

    with tokenizer.as_target_tokenizer():
        labels = tokenizer(examples["eval_data"], max_length=128,
truncation=True, padding="max_length")
```

```
    model_inputs["labels"] = labels["input_ids"]
    return model_inputs

# Convert pandas DataFrame to Hugging Face Dataset
hf_dataset = Dataset.from_pandas(dataset)
tokenized_dataset = hf_dataset.map(preprocess_function, batched=True)

# training arguments
training_args = TrainingArguments(
    output_dir="./results",
    num_train_epochs=3,
    per_device_train_batch_size=4,
    per_device_eval_batch_size=4,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir="./logs",
)

# Initialize Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset,
)

trainer.train()
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89:
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 Colab
and restart your session.
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(

tokenizer_config.json:   0%|          | 0.00/2.32k [00:00<?, ?B/s]

spiece.model:   0%|          | 0.00/792k [00:00<?, ?B/s]

tokenizer.json:   0%|          | 0.00/1.39M [00:00<?, ?B/s]

config.json:   0%|          | 0.00/1.21k [00:00<?, ?B/s]

model.safetensors:   0%|          | 0.00/242M [00:00<?, ?B/s]
```

```
generation_config.json:   0%|          | 0.00/147 [00:00<?, ?B/s]

Map:   0%|          | 0/19 [00:00<?, ? examples/s]

/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py:4126:
UserWarning: `as_target_tokenizer` is deprecated and will be removed in v5 of
Transformers. You can tokenize your labels by using the argument `text_target` of
the regular `__call__` method (either in the same call as your input texts if you
use the same keyword arguments, or in a separate call.
  warnings.warn(
wandb: WARNING The `run_name` is currently set to the same value as
`TrainingArguments.output_dir`. If this was not intended, please specify a
different run name by setting the `TrainingArguments.run_name` parameter.
wandb: Using wandb-core as the SDK backend. Please refer to
https://wandb.me/wandb-core for more information.

<IPython.core.display.Javascript object>

wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally:
https://wandb.me/wandb-server)
wandb: You can find your API key in your browser here: https://wandb.ai/authorize
wandb: Paste an API key from your profile and hit enter, or press ctrl+c to quit:

 ..........

wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

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

Step	Training Loss

```
TrainOutput(global_step=15, training_loss=10.728704833984375,
metrics={'train_runtime': 577.082, 'train_samples_per_second': 0.099,
'train_steps_per_second': 0.026, 'total_flos': 7714482683904.0, 'train_loss':
10.728704833984375, 'epoch': 3.0})
```

Explanations of our Neural Network weights and Biases

In this project, weights and biases are key in helping the model understand and process older language patterns. By adjusting the weights, the model learns and adapts to different linguistic structures and vocabularies that are not like modern English. During training, these weights are fine-tuned to reduce errors between the model's output and well-written summaries. Adjusting biases also helps the model avoid leaning too heavily on certain patterns, ensuring it can transform language consistently across different types of text.

```
# Load pre-trained model and tokenizer
model_name = "t5-small"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)

def summarize_and_modernize(text):
    inputs = tokenizer("summarize: " + text, return_tensors="pt", max_length=512,
truncation=True)
    summary_ids = model.generate(inputs["input_ids"], max_length=150,
min_length=40, length_penalty=2.0, num_beams=4, early_stopping=True)
    summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
    return summary

# Collect input texts from user or use defaults
def get_texts():
    texts = {}
    use_hardcoded = input("Would you like to enter custom texts? (yes/no):
").strip().lower() == "no"

    if use_hardcoded:
        # Use predefined hardcoded texts
        texts = {
            "Paradise Lost": "Of Rebel Angels, by whose aid aspiring To set
            himself in Glory above his Peers, He trusted to have equal'd the most
            High, If he oppos'd; and with ambitious aim Against the Throne and
            Monarchy of God Rais'd impious War in Heav'n and Battel proud With
            vain attempt.",
            "Hamlet": "Good Hamlet, cast thy nighted color off, And let thine eye
            look like a friend on Denmark. Do not forever with thy vailèd lids
            Seek for thy noble father in the dust. Thou know'st 'tis common; all
            that lives must die, Passing through nature to eternity.",
            "Macbeth": "Hath not yet lighted, and our safest way Is to avoid the
            aim. Therefore to horse, And let us not be dainty of leave-taking But
            shift away. There's warrant in that theft Which steals itself when
            there's no mercy left.",
            "Great Expectations": "Heaven knows we need never be ashamed of our
            tears, for they are rain upon the blinding dust of earth, overlying
            our hard hearts. I was better after I had cried, than before-more
            sorry, more aware of my own ingratitude, more gentle.",
```

```
            "Frankenstein": "For this I had deprived myself of rest and health. I
            had desired it with an ardour that far exceeded moderation; but now
            that I had finished, the beauty of the dream vanished, and breathless
            horror and disgust filled my heart."
        }
    else:
        # Collect custom texts
        print("Enter historical texts for modernization. Type 'done' when
        finished.")
        while True:
            title = input("Enter the title of the text (or type 'done' to
finish): ")
            if title.lower() == "done":
                break
            text = input(f"Enter the text for '{title}': ")
            texts[title] = text
    return texts

# Get texts from user or defaults
texts = get_texts()

# Generate modernized versions of the texts
modernized_texts = {k: summarize_and_modernize(v) for k, v in texts.items()}

# Display the modernized texts
for title, original in texts.items():
    modernized = modernized_texts[title]
    print(f"\nOriginal {title}:")
    print(original)
    print("\nModernized Version:")
    print(modernized)
    print("\n")
```

```
Would you like to enter custom texts? (yes/no): no

Original Paradise Lost:
Of Rebel Angels, by whose aid aspiring To set himself in Glory above his Peers,
He trusted to have equal'd the most High, If he oppos'd; and with ambitious aim
Against the Throne and Monarchy of God Rais'd impious War in Heav'n and Battel
proud With vain attempt.

Modernized Version:
Against the Throne and Monarchy of God Rais'd impious War in Heav'n and Battel.
Against the Throne and Monarchy of God Rais'd impious War.
```

Original Hamlet:
Good Hamlet, cast thy nighted color off, And let thine eye look like a friend on Denmark. Do not forever with thy vailèd lids Seek for thy noble father in the dust. Thou know'st 'tis common; all that lives must die, Passing through nature to eternity.

Modernized Version:
do not forever with thy vailèd lids Seek for thy noble father in the dust. all that lives must die, Passing through nature to eternity.

Original Macbeth:
Hath not yet lighted, and our safest way Is to avoid the aim. Therefore to horse, And let us not be dainty of leave-taking But shift away. There's warrant in that theft Which steals itself when there's no mercy left.

Modernized Version:
there's warrant in that theft which steals itself when there's no mercy left. there's warrant in that theft which steals itself when there's no mercy left.

Original Great Expectations:
Heaven knows we need never be ashamed of our tears, for they are rain upon the blinding dust of earth, overlying our hard hearts. I was better after I had cried, than before-more sorry, more aware of my own ingratitude, more gentle.

Modernized Version:
tears are rain upon the blinding dust of earth, overlying our hard hearts. Heaven knows we need never be ashamed of our tears, for they are rain upon the blinding dust of earth.

Original Frankenstein:
For this I had deprived myself of rest and health. I had desired it with an ardour that far exceeded moderation; but now that I had finished, the beauty of the dream vanished, and breathless horror and disgust filled my heart.

Modernized Version:
for this I had deprived myself of rest and health. now that I had finished, the beauty of the dream vanished. breathless horror filled my heart with disgust and horror.

Instructions for use for the Client

- Before usage, the user must install the following packages: datasets, transformers, pandas, torch, and google colab
- The user also must have an API key ready when asked for one in order to run the training for the model. They have to follow a provided link to create an account to get the API key.
- The user simply needs to run the code and insert the desired text into the text box and the model will return a summarized and modernized version of the text
- If the user just wants to test the model but doesn't have anything to enter at the moment, there are built in test cases for the user to analyze to see if they like how it is working.

Conclusion

This text summarization project demonstrates the application of deep learning techniques to automate the process of condensing lengthy texts into concise summaries. By utilizing a custom dataset of old British texts and their professional summaries, the model is tailored to handle specific language styles and structures.

The encoder-decoder model with attention mechanisms is useful for creating summaries, as it helps the model focus on important parts of the input text. Using LSTM layers allows the model to remember information over longer sections of text, which is essential for making accurate summaries.

To improve this model, we could add more data or adjust the settings (like learning rate or layer size). Testing the model with different types of texts and comparing its summaries to human-made ones would help us see how well it works and where it might need adjustments.