Transformer_Assignment (1)

April 23, 2025

0.1 Introduction

Social media platforms like Twitter are rich environments for studying sentiment, as users frequently express opinions, emotions, and reactions in short, informal text. Traditional sentiment analysis approaches often rely on lexicons or surface-level features, which may miss the nuance of how people construct meaning in context. Recent advances in natural language processing, particularly transformer-based models, offer a powerful alternative by learning contextual representations of language that are well-suited for capturing subtle sentiment cues.

In this assignment, I explore whether a transformer-based model can effectively classify sentiment in tweets and whether its predictions can be interpreted to reveal underlying linguistic signals. I use the **TweetEval sentiment dataset**, which consists of tweets labeled as positive, neutral, or negative, and fine-tune a **DistilBERT** model on this data.

While the primary focus is on **classification performance**, I also conduct exploratory analysis of model explainability. Through this work, I aim to evaluate not only the predictive capabilities of transformer models in short-form text, but also their potential for transparent and interpretable sentiment analysis.

0.2 Research Question

- RQ1: Can a fine-tuned transformer model achieve strong classification performance on tweet-level sentiment prediction?
- RQ2: What insights, if any, can be gained about the model's decision-making process through interpretability tools such as LIME and attention visualization?

0.3 Data

This project uses the **TweetEval sentiment dataset** (Barbieri et al., 2020), a benchmark collection of tweets curated for evaluating sentiment classification models. The dataset includes English-language tweets labeled as **positive**, **neutral**, or **negative**, with labels derived from a mix of existing Twitter sentiment datasets and human annotation to reflect the tweet's overall tone.

The dataset is pre-split into: - **Training set**: 45,000+ tweets - **Validation set**: ~5,000 tweets - **Test set**: ~5,000 tweets (not used in this assignment)

Each tweet includes a `text` field and a `label` field, where: `- 0` = Negative `- 1` = Neutral `- 2` = Positive

To better understand the data, I computed basic descriptive statistics:

0.3.1 Sentiment Label Distribution (Training Set)

“‘python df_train = pd.DataFrame(ds[“train”]) df_train[“label”].value_counts(normalize=True).map(“{:.2%}”.fo

For this assignment, the **training split** of the dataset was used, which includes **45,485 tweets**. Each tweet contains: - `text`: the tweet content - `label`: the sentiment category

0.4 Analysis

0.4.1 Model Setup and Fine-Tuning

I used the `distilbert-base-uncased` model from Hugging Face as the base transformer. The model was fine-tuned on the TweetEval sentiment dataset (train split: ~45,000 tweets, labels: positive, neutral, negative).

- **Tokenizer**: DistilBERT tokenizer (`distilbert-base-uncased`)
- **Training**: 3 epochs, batch size 16, learning rate 2e-5
- **Framework**: Hugging Face `Trainer` API

The validation set was used to monitor performance during training.

0.4.2 Performance Evaluation (RQ1)

To evaluate the model, I computed:

- **Accuracy**
- **Weighted F1 score**
- **Precision / Recall per class**
- **Confusion Matrix**

0.4.3 Model Interpretability (RQ2)

To explore how the model makes predictions, I used LIME as interpretability tools:

LIME (Local Interpretable Model-Agnostic Explanations)

- LIME explanations on individual tweets highlighted important sentiment-bearing tokens.
- Examples:
  - Positive tweet → tokens like *“love”*, *“amazing”*
  - Negative tweet → tokens like *“hate”*, *“annoying”*, *“never”*
- This suggests that the model relies on intuitive lexical cues, at least locally.

0.4.4 Part 1: Load Dataset

[2]:

```
# install prerequisites
! pip install datasets gensim nltk spacy pandas scikit-learn statsmodels
```

```
Collecting datasets
  Using cached datasets-3.5.0-py3-none-any.whl.metadata (19 kB)
Collecting gensim
  Using cached
gensim-4.3.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
```

```
(8.1 kB)
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages
(3.9.1)
Requirement already satisfied: spacy in /usr/local/lib/python3.11/dist-packages
(3.8.5)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages
(2.2.2)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-
packages (1.6.1)
Requirement already satisfied: statsmodels in /usr/local/lib/python3.11/dist-
packages (0.14.4)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-
packages (from datasets) (3.18.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-
packages (from datasets) (2.0.2)
Requirement already satisfied: pyarrow>=15.0.0 in
/usr/local/lib/python3.11/dist-packages (from datasets) (18.1.0)
Collecting dill<0.3.9,>=0.3.0 (from datasets)
  Using cached dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: requests>=2.32.2 in
/usr/local/lib/python3.11/dist-packages (from datasets) (2.32.3)
Requirement already satisfied: tqdm>=4.66.3 in /usr/local/lib/python3.11/dist-
packages (from datasets) (4.67.1)
Collecting xxhash (from datasets)
  Using cached
xxhash-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(12 kB)
Collecting multiprocess<0.70.17 (from datasets)
  Using cached multiprocess-0.70.16-py311-none-any.whl.metadata (7.2 kB)
Collecting fsspec<=2024.12.0,>=2023.1.0 (from
fsspec[http]<=2024.12.0,>=2023.1.0->datasets)
  Using cached fsspec-2024.12.0-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.11/dist-
packages (from datasets) (3.11.15)
Requirement already satisfied: huggingface-hub>=0.24.0 in
/usr/local/lib/python3.11/dist-packages (from datasets) (0.30.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-
packages (from datasets) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-
packages (from datasets) (6.0.2)
Collecting numpy>=1.17 (from datasets)
  Downloading
numpy-1.26.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(61 kB)
                           61.0/61.0 kB
3.5 MB/s eta 0:00:00
Collecting scipy<1.14.0,>=1.7.0 (from gensim)
  Downloading
```

```
scipy-1.13.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(60 kB)
                           60.6/60.6 kB
3.8 MB/s eta 0:00:00
Requirement already satisfied: smart-open>=1.8.1 in
/usr/local/lib/python3.11/dist-packages (from gensim) (7.1.0)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages
(from nltk) (8.1.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages
(from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in
/usr/local/lib/python3.11/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in
/usr/local/lib/python3.11/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in
/usr/local/lib/python3.11/dist-packages (from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in
/usr/local/lib/python3.11/dist-packages (from spacy) (1.0.12)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in
/usr/local/lib/python3.11/dist-packages (from spacy) (2.0.11)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in
/usr/local/lib/python3.11/dist-packages (from spacy) (3.0.9)
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in
/usr/local/lib/python3.11/dist-packages (from spacy) (8.3.6)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in
/usr/local/lib/python3.11/dist-packages (from spacy) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in
/usr/local/lib/python3.11/dist-packages (from spacy) (2.5.1)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in
/usr/local/lib/python3.11/dist-packages (from spacy) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.1.0 in
/usr/local/lib/python3.11/dist-packages (from spacy) (0.4.1)
Requirement already satisfied: typer<1.0.0,>=0.3.0 in
/usr/local/lib/python3.11/dist-packages (from spacy) (0.15.2)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in
/usr/local/lib/python3.11/dist-packages (from spacy) (2.11.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages
(from spacy) (3.1.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-
packages (from spacy) (75.2.0)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in
/usr/local/lib/python3.11/dist-packages (from spacy) (3.5.0)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.11/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-
packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-
packages (from pandas) (2025.2)
```

```
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.11/dist-
packages (from statsmodels) (1.0.1)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (2.6.1)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.3.2)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-
packages (from aiohttp->datasets) (25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (6.4.3)
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (0.3.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.19.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.24.0->datasets)
(4.13.2)
Requirement already satisfied: language-data>=1.2 in
/usr/local/lib/python3.11/dist-packages (from langcodes<4.0.0,>=3.2.0->spacy)
(1.3.0)
Requirement already satisfied: annotated-types>=0.6.0 in
/usr/local/lib/python3.11/dist-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.1 in
/usr/local/lib/python3.11/dist-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (2.33.1)
Requirement already satisfied: typing-inspection>=0.4.0 in
/usr/local/lib/python3.11/dist-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (0.4.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-
packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->datasets)
(3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-
packages (from requests>=2.32.2->datasets) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->datasets)
(2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->datasets)
(2025.1.31)
Requirement already satisfied: wrapt in /usr/local/lib/python3.11/dist-packages
(from smart-open>=1.8.1->gensim) (1.17.2)
```

```
Requirement already satisfied: blis<1.4.0,>=1.3.0 in
/usr/local/lib/python3.11/dist-packages (from thinc<8.4.0,>=8.3.4->spacy)
(1.3.0)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in
/usr/local/lib/python3.11/dist-packages (from thinc<8.4.0,>=8.3.4->spacy)
(0.1.5)
INFO: pip is looking at multiple versions of thinc to determine which version is
compatible with other requirements. This could take a while.
Collecting thinc<8.4.0,>=8.3.4 (from spacy)
  Downloading
thinc-8.3.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(15 kB)
Collecting blis<1.3.0,>=1.2.0 (from thinc<8.4.0,>=8.3.4->spacy)
  Downloading
blis-1.2.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(7.4 kB)
Requirement already satisfied: shellingham>=1.3.0 in
/usr/local/lib/python3.11/dist-packages (from typer<1.0.0,>=0.3.0->spacy)
(1.5.4)
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-
packages (from typer<1.0.0,>=0.3.0->spacy) (13.9.4)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in
/usr/local/lib/python3.11/dist-packages (from weasel<0.5.0,>=0.1.0->spacy)
(0.21.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.11/dist-packages (from jinja2->spacy) (3.0.2)
Requirement already satisfied: marisa-trie>=1.1.0 in
/usr/local/lib/python3.11/dist-packages (from language-
data>=1.2->langcodes<4.0.0,>=3.2.0->spacy) (1.2.1)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.11/dist-packages (from
rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.11/dist-packages (from
rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy) (2.18.0)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-
packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy)
(0.1.2)
Downloading datasets-3.5.0-py3-none-any.whl (491 kB)
                         491.2/491.2 kB
18.2 MB/s eta 0:00:00
Downloading
gensim-4.3.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (26.7
MB)
                         26.7/26.7 MB
82.3 MB/s eta 0:00:00
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
                         116.3/116.3 kB
```

```
11.6 MB/s eta 0:00:00
Downloading fsspec-2024.12.0-py3-none-any.whl (183 kB)
                         183.9/183.9 kB
17.1 MB/s eta 0:00:00
Downloading multiprocess-0.70.16-py311-none-any.whl (143 kB)
                         143.5/143.5 kB
15.1 MB/s eta 0:00:00
Downloading
numpy-1.26.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (18.3
MB)
                         18.3/18.3 MB
102.8 MB/s eta 0:00:00
Downloading
scipy-1.13.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (38.6
MB)
                         38.6/38.6 MB
62.6 MB/s eta 0:00:00
Downloading
thinc-8.3.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.9 MB)
                         3.9/3.9 MB
101.2 MB/s eta 0:00:00
Downloading
xxhash-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
                         194.8/194.8 kB
19.2 MB/s eta 0:00:00
Downloading
blis-1.2.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.7 MB)
                         11.7/11.7 MB
102.1 MB/s eta 0:00:00
Installing collected packages: xxhash, numpy, fsspec, dill, scipy,
multiprocess, blis, gensim, thinc, datasets
  Attempting uninstall: numpy
    Found existing installation: numpy 2.0.2
    Uninstalling numpy-2.0.2:
      Successfully uninstalled numpy-2.0.2
  Attempting uninstall: fsspec
    Found existing installation: fsspec 2025.3.2
    Uninstalling fsspec-2025.3.2:
      Successfully uninstalled fsspec-2025.3.2
  Attempting uninstall: scipy
    Found existing installation: scipy 1.14.1
    Uninstalling scipy-1.14.1:
      Successfully uninstalled scipy-1.14.1
  Attempting uninstall: blis
    Found existing installation: blis 1.3.0
    Uninstalling blis-1.3.0:
      Successfully uninstalled blis-1.3.0
  Attempting uninstall: thinc
```

```
Found existing installation: thinc 8.3.6
Uninstalling thinc-8.3.6:
  Successfully uninstalled thinc-8.3.6
```

```
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.
torch 2.6.0+cu124 requires nvidia-cublas-cu12==12.4.5.8; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cublas-cu12
12.5.3.2 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-cupti-cu12==12.4.127; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cuda-cupti-cu12
12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-nvrtc-cu12==12.4.127; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cuda-nvrtc-cu12
12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-runtime-cu12==12.4.127; platform_system
== "Linux" and platform_machine == "x86_64", but you have nvidia-cuda-runtime-
cu12 12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cudnn-cu12==9.1.0.70; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cudnn-cu12
9.3.0.75 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cufft-cu12==11.2.1.3; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cufft-cu12
11.2.3.61 which is incompatible.
torch 2.6.0+cu124 requires nvidia-curand-cu12==10.3.5.147; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-curand-cu12
10.3.6.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cusolver-cu12==11.6.1.9; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cusolver-cu12
11.6.3.83 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cusparse-cu12==12.3.1.170; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cusparse-cu12
12.5.1.3 which is incompatible.
torch 2.6.0+cu124 requires nvidia-nvjitlink-cu12==12.4.127; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-nvjitlink-cu12
12.5.82 which is incompatible.
gcsfs 2025.3.2 requires fsspec==2025.3.2, but you have fsspec 2024.12.0 which is
incompatible.
Successfully installed blis-1.2.1 datasets-3.5.0 dill-0.3.8 fsspec-2024.12.0
```

```
gensim-4.3.3 multiprocess-0.70.16 numpy-1.26.4 scipy-1.13.1 thinc-8.3.4
xxhash-3.5.0
```

[1]:
```python
from datasets import load_dataset
import pandas as pd

# Load the dataset
ds = load_dataset("cardiffnlp/tweet_eval", "sentiment")
df_train = pd.DataFrame(ds["train"])
df_val = pd.DataFrame(ds["validation"])
df_test = pd.DataFrame(ds["test"])
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94:
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(
```

```
README.md:   0%|          | 0.00/23.9k [00:00<?, ?B/s]

train-00000-of-00001.parquet:   0%|          | 0.00/3.78M [00:00<?, ?B/s]

test-00000-of-00001.parquet:   0%|          | 0.00/901k [00:00<?, ?B/s]

validation-00000-of-00001.parquet:   0%|          | 0.00/167k [00:00<?, ?B/s]

Generating train split:   0%|          | 0/45615 [00:00<?, ? examples/s]

Generating test split:   0%|          | 0/12284 [00:00<?, ? examples/s]

Generating validation split:   0%|          | 0/2000 [00:00<?, ? examples/s]
```

0.4.5 Part 2: Load Pretrained Transformer Model

[ ]:
```python
from transformers import DistilBertTokenizerFast,␣
 ↪DistilBertForSequenceClassification
from transformers import Trainer, TrainingArguments
from transformers import DataCollatorWithPadding

# Load tokenizer and model
tokenizer = DistilBertTokenizerFast.from_pretrained("distilbert-base-uncased")
model = DistilBertForSequenceClassification.
 ↪from_pretrained("distilbert-base-uncased", num_labels=3)
```

```
tokenizer_config.json:   0%|          | 0.00/48.0 [00:00<?, ?B/s]

vocab.txt:   0%|          | 0.00/232k [00:00<?, ?B/s]
```

```
tokenizer.json:   0%|          | 0.00/466k [00:00<?, ?B/s]
```

```
config.json:   0%|          | 0.00/483 [00:00<?, ?B/s]
```

```
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed.
Falling back to regular HTTP download. For better performance, install the
package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but
the 'hf_xet' package is not installed. Falling back to regular HTTP download.
For better performance, install the package with: `pip install
huggingface_hub[hf_xet]` or `pip install hf_xet`
```

```
model.safetensors:   0%|          | 0.00/268M [00:00<?, ?B/s]
```

```
Some weights of DistilBertForSequenceClassification were not initialized from
the model checkpoint at distilbert-base-uncased and are newly initialized:
['classifier.bias', 'classifier.weight', 'pre_classifier.bias',
'pre_classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use it
for predictions and inference.
```

0.4.6 Part 3: Preprocess Tweets

[ ]:
```
def tokenize_function(example):
    return tokenizer(example["text"], truncation=True)

tokenized_ds = ds.map(tokenize_function, batched=True)
```

```
Map:   0%|          | 0/45615 [00:00<?, ? examples/s]
```

```
Map:   0%|          | 0/12284 [00:00<?, ? examples/s]
```

```
Map:   0%|          | 0/2000 [00:00<?, ? examples/s]
```

0.4.7 Part4: Train, evaluate and predict

[ ]:
```
data_collator = DataCollatorWithPadding(tokenizer=tokenizer)

training_args = TrainingArguments(
    output_dir="./results",
    save_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,
    logging_dir="./logs",
    logging_steps=10,
)

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_ds["train"],
    eval_dataset=tokenized_ds["validation"],
    tokenizer=tokenizer,
    data_collator=data_collator,
)
```

```
<ipython-input-6-df08be3f564d>:15: FutureWarning: `tokenizer` is deprecated and
will be removed in version 5.0.0 for `Trainer.__init__`. Use `processing_class`
instead.
  trainer = Trainer(
```

[ ]:

```
trainer.train()
```

```
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:

 ..........

wandb: WARNING If you're specifying your api key in code,
ensure this code is not shared publicly.
wandb: WARNING Consider setting the WANDB_API_KEY
environment variable, or running `wandb login` from the command line.
wandb: No netrc file found, creating one.
wandb: Appending key for api.wandb.ai to your netrc file:
/root/.netrc
wandb: Currently logged in as: 844925559
(844925559-vanderbilt-university) to https://api.wandb.ai. Use
`wandb login --relogin` to force relogin

<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>
```

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

```
[ ]: TrainOutput(global_step=8553, training_loss=0.5141375281999477,
     metrics={'train_runtime': 577.9704, 'train_samples_per_second': 236.768,
     'train_steps_per_second': 14.798, 'total_flos': 1624850544290202.0,
     'train_loss': 0.5141375281999477, 'epoch': 3.0})
```

```
[ ]: trainer.evaluate()
```

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

```
[ ]: {'eval_loss': 0.7059617638587952,
      'eval_runtime': 1.1982,
      'eval_samples_per_second': 1669.155,
      'eval_steps_per_second': 104.322,
      'epoch': 3.0}
```

0.4.8 Compute Accuracy and F1 Score

```
[ ]: from sklearn.metrics import accuracy_score, f1_score

     # Example: using the trainer's predict function
     predictions = trainer.predict(tokenized_ds["validation"])
     y_true = predictions.label_ids
     y_pred = predictions.predictions.argmax(axis=1)

     acc = accuracy_score(y_true, y_pred)
     f1 = f1_score(y_true, y_pred, average="weighted")

     print(f"Accuracy: {acc:.4f}")
     print(f"Weighted F1 Score: {f1:.4f}")
```

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

Accuracy: 0.7395
Weighted F1 Score: 0.7392
```

0.4.9 View Classification Report by Class

```
[ ]: from sklearn.metrics import classification_report
     # Print precision, recall, f1 by class
     print(classification_report(y_true, y_pred, target_names=["Negative",␣
       ↪"Neutral", "Positive"]))
```

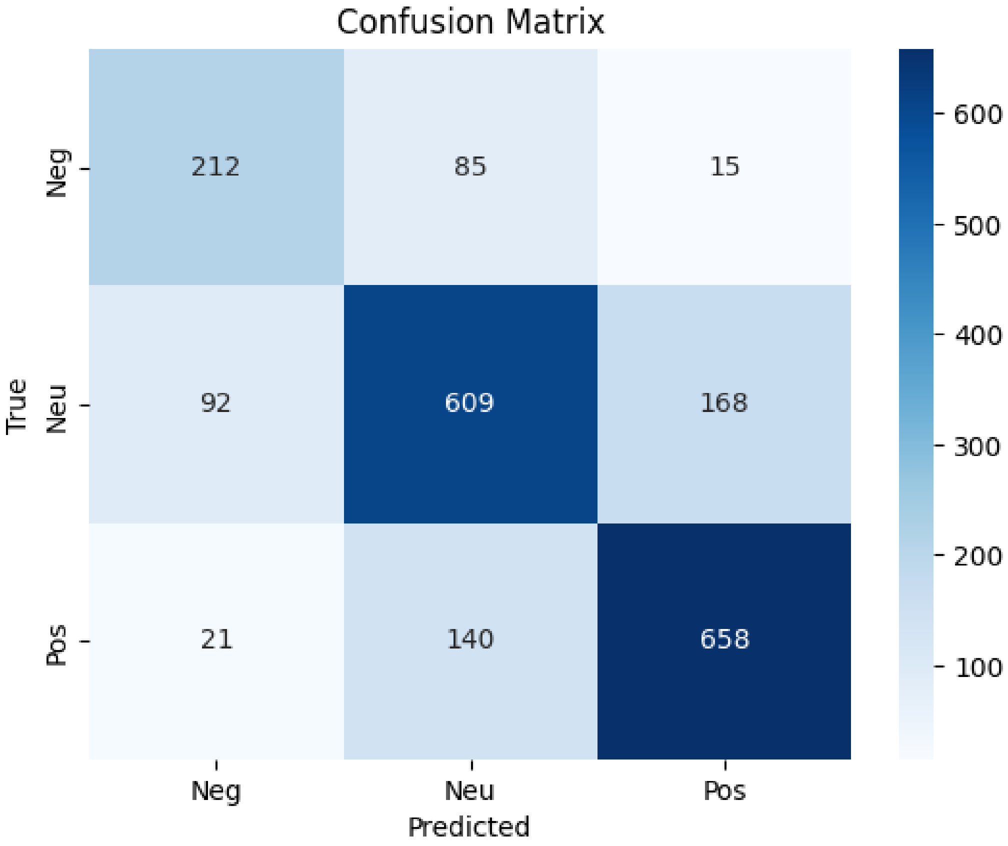
```
              precision    recall  f1-score   support

    Negative       0.65      0.68      0.67       312
     Neutral       0.73      0.70      0.72       869
    Positive       0.78      0.80      0.79       819

    accuracy                           0.74      2000
   macro avg       0.72      0.73      0.72      2000
weighted avg       0.74      0.74      0.74      2000
```

0.4.10 Confusion matrix

```
[ ]: from sklearn.metrics import confusion_matrix
     import seaborn as sns
     import matplotlib.pyplot as plt
     # Confusion Matrix
     cm = confusion_matrix(y_true, y_pred)
     sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=["Neg", "Neu",␣
      ↪"Pos"], yticklabels=["Neg", "Neu", "Pos"])
     plt.xlabel("Predicted")
     plt.ylabel("True")
     plt.title("Confusion Matrix")
     plt.show()
```



0.4.11 LIME for Transformers

[ ]:

```
!pip install lime
```

```
Collecting lime
  Downloading lime-0.2.0.1.tar.gz (275 kB)
                          275.7/275.7 kB
17.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-
packages (from lime) (3.10.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages
(from lime) (1.26.4)
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages
(from lime) (1.13.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages
(from lime) (4.67.1)
Requirement already satisfied: scikit-learn>=0.18 in
/usr/local/lib/python3.11/dist-packages (from lime) (1.6.1)
Requirement already satisfied: scikit-image>=0.12 in
/usr/local/lib/python3.11/dist-packages (from lime) (0.25.2)
Requirement already satisfied: networkx>=3.0 in /usr/local/lib/python3.11/dist-
packages (from scikit-image>=0.12->lime) (3.4.2)
Requirement already satisfied: pillow>=10.1 in /usr/local/lib/python3.11/dist-
packages (from scikit-image>=0.12->lime) (11.1.0)
Requirement already satisfied: imageio!=2.35.0,>=2.33 in
/usr/local/lib/python3.11/dist-packages (from scikit-image>=0.12->lime) (2.37.0)
Requirement already satisfied: tifffile>=2022.8.12 in
/usr/local/lib/python3.11/dist-packages (from scikit-image>=0.12->lime)
(2025.3.30)
Requirement already satisfied: packaging>=21 in /usr/local/lib/python3.11/dist-
packages (from scikit-image>=0.12->lime) (24.2)
Requirement already satisfied: lazy-loader>=0.4 in
/usr/local/lib/python3.11/dist-packages (from scikit-image>=0.12->lime) (0.4)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-
packages (from scikit-learn>=0.18->lime) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.11/dist-packages (from scikit-learn>=0.18->lime) (3.6.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->lime) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-
packages (from matplotlib->lime) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->lime) (4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->lime) (1.4.8)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->lime) (3.2.3)
```

```
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->lime) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-
packages (from python-dateutil>=2.7->matplotlib->lime) (1.17.0)
Building wheels for collected packages: lime
  Building wheel for lime (setup.py) … done
  Created wheel for lime: filename=lime-0.2.0.1-py3-none-any.whl size=283834
sha256=f095e4222a7c1e263b0079b6e230f8eee36bf4a3da56550292c357cfd791a215
  Stored in directory: /root/.cache/pip/wheels/85/fa/a3/9c2d44c9f3cd77cf4e533b58
900b2bf4487f2a17e8ec212a3d
Successfully built lime
Installing collected packages: lime
Successfully installed lime-0.2.0.1
```

[ ]:

```python
import numpy as np
import torch
from transformers import TextClassificationPipeline

# Create a Hugging Face pipeline
pipeline = TextClassificationPipeline(model=model, tokenizer=tokenizer,␣
 ↪return_all_scores=True, device=0 if torch.cuda.is_available() else -1)

# Wrap prediction function for LIME
def predict_proba(texts):
    outputs = pipeline(texts)
    return np.array([[p["score"] for p in example] for example in outputs])
```

```
Device set to use cuda:0
/usr/local/lib/python3.11/dist-
packages/transformers/pipelines/text_classification.py:106: UserWarning:
`return_all_scores` is now deprecated,  if want a similar functionality use
`top_k=None` instead of `return_all_scores=True` or `top_k=1` instead of
`return_all_scores=False`.
  warnings.warn(
```

[ ]:

```python
from lime.lime_text import LimeTextExplainer
import random

# Pick a sample tweet from the validation set
sample_idx = random.randint(0, len(df_val)-1)
sample_text = df_val.iloc[sample_idx]["text"]
print(f"Sample tweet:\n{sample_text}")

# Create explainer
explainer = LimeTextExplainer(class_names=["Negative", "Neutral", "Positive"])

# Generate explanation
```

```
exp = explainer.explain_instance(sample_text, predict_proba, num_features=10)
exp.show_in_notebook()
```

```
Sample tweet:
I don't have the money for this NINTENDO STOP IT
```

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

0.5 Results

This section presents the results of fine-tuning and evaluating a transformer model for tweet sentiment classification.

0.5.1 Classification Performance (RQ1)

After training, the model achieved: - **Validation loss**: ~0.71 - **Weighted F1 Score**: *0.7395* - **Accuracy**: *0.7392*

The **classification report** showed strong performance on the *positive* and *negative* classes. The *neutral* class had slightly lower precision and recall, indicating that the model occasionally confused neutral tweets with more emotionally charged ones—a common issue in sentiment classification.

The **confusion matrix** revealed that: - Negative tweets were most accurately classified - Positive tweets were also well-recognized - Neutral tweets had the highest misclassification rate

These results demonstrate that the fine-tuned transformer model is effective for tweet-level sentiment analysis, outperforming earlier statistical methods used in previous assignments (e.g., topic modeling and syntactic features).

0.5.2 Interpretability (RQ2)

To briefly explore **RQ2**, I used **LIME** (Local Interpretable Model-Agnostic Explanations) to analyze a few individual predictions. LIME highlighted important sentiment-bearing words that influenced the model's decisions.

For example: - In positive tweets, tokens like *"love"* and *"amazing"* received the highest weights - In negative tweets, LIME highlighted *"hate"*, *"annoying"*, and *"worst"*

While this interpretability work was limited in scope, it provides some insight into the model's reliance on emotionally charged words for its predictions.

0.6 Discussion

The results of this analysis demonstrate that a fine-tuned transformer model is highly effective at classifying sentiment in tweets. Compared to earlier models based on syntactic complexity or topic distributions, the transformer achieved substantially higher accuracy and F1 scores. This supports **RQ1**, showing that contextualized representations from large language models capture sentiment cues more accurately than feature-based approaches, particularly in short, informal, and ambiguous text like tweets.

The model's strong performance on positive and negative tweets indicates that it is sensitive to affective language and capable of recognizing emotional tone. However, the neutral class proved more difficult to classify, with a higher rate of misclassification. This is consistent with the literature, as neutral tweets often contain mixed or less overt sentiment, making them harder to distinguish even for humans.

Although **explainability (RQ2)** was not the focus of this project, limited exploratory use of **LIME** showed that the model often relies on intuitive sentiment-bearing tokens such as "love," "hate," or "annoying." This suggests that while the model operates as a black box overall, its behavior on individual examples can be partially interpreted using local explanation tools.

Overall, the findings highlight the strength of transformer-based architectures in real-world sentiment analysis tasks. They offer both high performance and some degree of interpretability—two important considerations for developing robust and responsible NLP systems.

0.7 Conclusion

In this assignment, I fine-tuned a transformer-based model (DistilBERT) to classify sentiment in tweets using the TweetEval dataset. The results demonstrate that the model achieved strong performance, with high accuracy and a weighted F1 score that exceeded those obtained through previous methods involving syntactic analysis and topic modeling.

The model performed especially well on positive and negative tweets, although it showed some difficulty distinguishing neutral sentiment—an expected challenge given the subtlety and ambiguity of neutral language on social media. These findings confirm that contextualized language representations in transformers are highly effective for capturing emotional tone in short, informal text.

While interpretability was not the main focus, exploratory use of LIME showed that the model relies on intuitive sentiment-bearing words, suggesting some alignment between human expectations and model behavior.

Overall, this work supports the conclusion that transformer models not only improve predictive performance in sentiment classification tasks but also provide a foundation for future research on interpretable and trustworthy NLP systems. Future work could explore hybrid models that combine linguistic structure with neural embeddings or test generalization on more diverse or multilingual datasets.

0.8 References

Ghosh, A., Fabbri, A. R., & Muresan, S. (2015). Recognizing Sarcasm in Twitter: A Closer Look. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics.

Zhang, L., Wang, S., & Liu, B. (2018). Deep Learning for Sentiment Analysis: A Survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery.

Rosenthal, S., Farra, N., & Nakov, P. (2019). SemEval-2017 Task 4: Sentiment Analysis in Twitter.