

CLASSIFICATION OF AI VS HUMAN WRITTEN ESSAYS

Ziqi Zhao, CSCI S-89, Final Project Report

I. Abstract

In the rapidly evolving field of generative AI, distinguishing AI-generated text from human-written content remains a critical challenge with implications in academia, content moderation, and digital content creation.

In January 2023, OpenAI had released an AI classifier for indicating AI-written text. However, it was shut down in July 2023 due to low accuracy rates¹. In the words of OpenAI:

"Our classifier is not fully reliable. In our evaluations on a "challenge set" of English texts, our classifier correctly identifies 26% of AI-written text (true positives) as "likely AI-written," while incorrectly labeling human-written text as AI-written 9% of the time (false positives). Our classifier's reliability typically improves as the length of the input text increases. Compared to our previously released classifier(opens in a new window), this new classifier is significantly more reliable on text from more recent AI systems."

This project proposes to use supervised deep learning techniques to build a classifier that attempts to classify AI vs human produced texts.

A DAIGTv2 dataset using more than 15,000 AI-written essays and more than 25,000 pre-college student human written essays were used for training and validation.

Specifically, an LSTM, simple RNN, and Bidirectional LSTM was trained for the classification task, with dropout layers included to prevent overfitting. These models achieved a 97-99% accuracy on validation data, compared to a benchmark simple logistic regression that only achieved a 64% accuracy.

A simple application was also created with a textbox that can be used to input text, which then shows a classification of whether the text was written by AI or human and a confidence percentage associated with the classification. Individual in-sample and out-of-sample prompts were tested for analysis.

A preliminary conclusion was reached that the "Scaling Law" of increased data resulting in increased accuracy and robustness should hold for this use case given more data, specifically to decrease false negative rates (AI-written incorrectly classified as human written). Synthetic data may be used for these purposes going forward. All code and the simple application can be found in the Jupyter notebooks/appendix.

¹ <https://openai.com/index/new-ai-classifier-for-indicating-ai-written-text/>

II. Background and Initial Exploratory Data Analysis (EDA)

There are many datasets available that collect human vs AI generated data. Upon exploration, I have chosen the DAIGT V2 Train Dataset2. The dataset employed in this project comprises approximately 44,868 essays compiled in one single .csv file, roughly equally distributed between AI-generated and human-written texts.

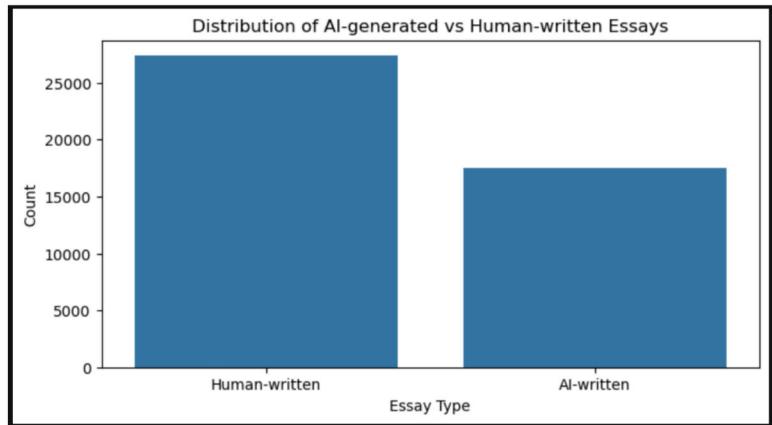


Figure 1. Distribution of AI-written vs Human-written Essays

Each essay is associated with the original metadata shown below including its label (1 for AI and 0 for human), prompt, source dataset, and length. This dataset's balanced nature and rich metadata makes it ideal for training and subsequent validation.

AI-generated essays:			
25996	In recent years, technology has had a profoun...	text	label \
25997	I strongly believe that meditation and mindful...		1
25998	One way school administrators can attempt to c...		1
25999	While summer is meant as a break from the regu...		1
26000	The use of Facial Action Coding System (FACS) ...		1
		prompt_name	RDizzl3_seven
25996	Car-free cities	mistral7binstruct_v2	True
25997	Distance learning	llama_70b_v1	False
25998	Cell phones at school	chat_gpt_moth	False
25999	Summer projects	darragh_claude_v7	False
26000	Facial action coding system	darragh_claude_v6	True

(Side Note: RDizzl3_seven feature — This is a boolean value feature present in the DAIGT-V2 dataset that determines whether essays were written based on prompts included in the hidden test set of the original Kaggle competition that this dataset was originally used for. We will ignore this column completely in this project.)

2 <https://www.kaggle.com/datasets/thedrcat/daijt-v2-train-dataset>

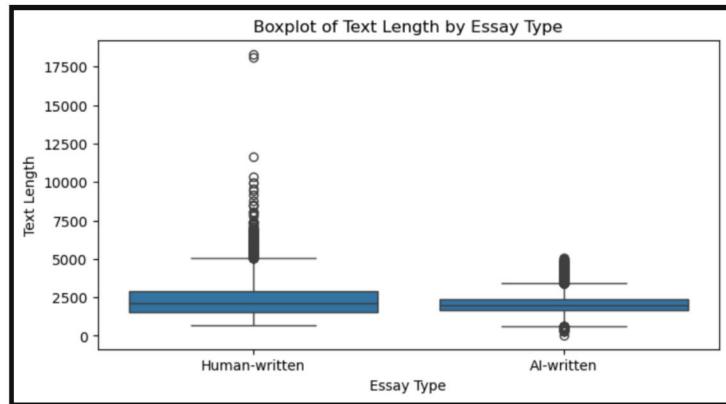


Figure 2. Boxplot of AI vs human written essay length.
The longer the essay length the more likely it was written by a human.

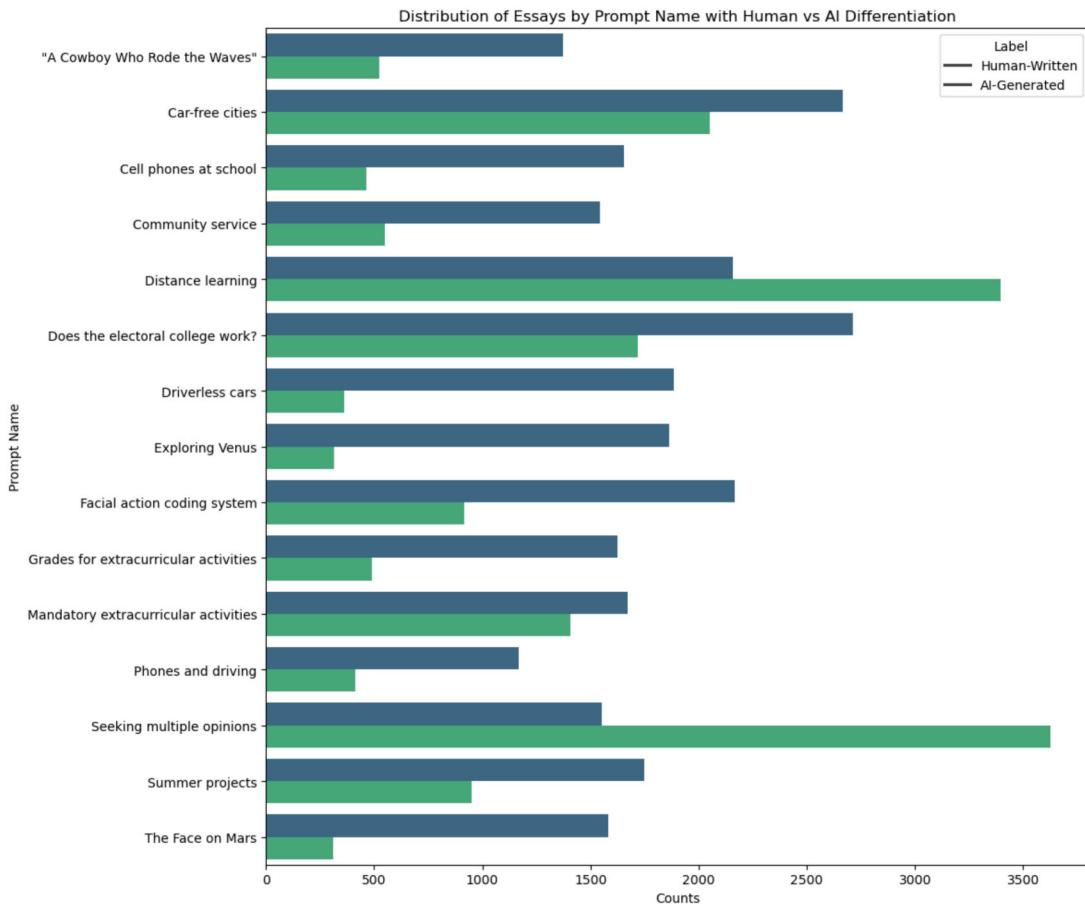


Figure 3. There are 15 different prompts used in essay topic, with some having more human-written essays and some more AI-written essays.

Figure 2 - The lengths of essays in the dataset were analyzed and the average and outlier essay lengths are shown. It is common and (currently) widely accepted that the longer the essay length the more likely it is to be written by a human.

Figure 3 - Human-written essays were taken from the Persuasive Essays for Rating, Selecting, and Understanding Argumentative and Discourse Elements (PERSUADE) corpus. The PERSUADE corpus is large-scale corpus of writing with annotated discourse elements³. The below is a summary of the PERSUADE corpus used:

“The PERSUADE corpus comprises two sub-corpora consisting of source-based essays ($n = 12,875$) and independent essays ($n = 13,121$). Source-based writing requires the student to refer to a text while independent writing excludes this requirement. The source-based set was derived from seven unique writing prompts and related sources. The writing reflects students in grades 6 through 10. The independent set reflects writing where background knowledge of the topic was not a requirement, and no sources were required to produce the texts. The independent sub-corpus was collected from students in grades 8 through 12, and the collection was derived from eight unique writing prompts. All prompts and sources are available within the PERSUADE corpus.”

The PERSUADE corpus was limited to essays with a minimum of 150 words of which 75% had to be correctly spelled American English words.”

There are many types of large language models (“LLM”) represented in the dataset, for example:

- ChatGPT by openAI, the most well-known and most adopted LLM
- Llama, a mostly open source LLM lead by Meta
- Mistral, an LLM that uses a fairly new “Mixture of Experts” model architecture
- Cohere Command, an LLM that uses retrieval-augmented generation to better handle longer texts.

³ <https://www.sciencedirect.com/science/article/pii/S1075293522000630>

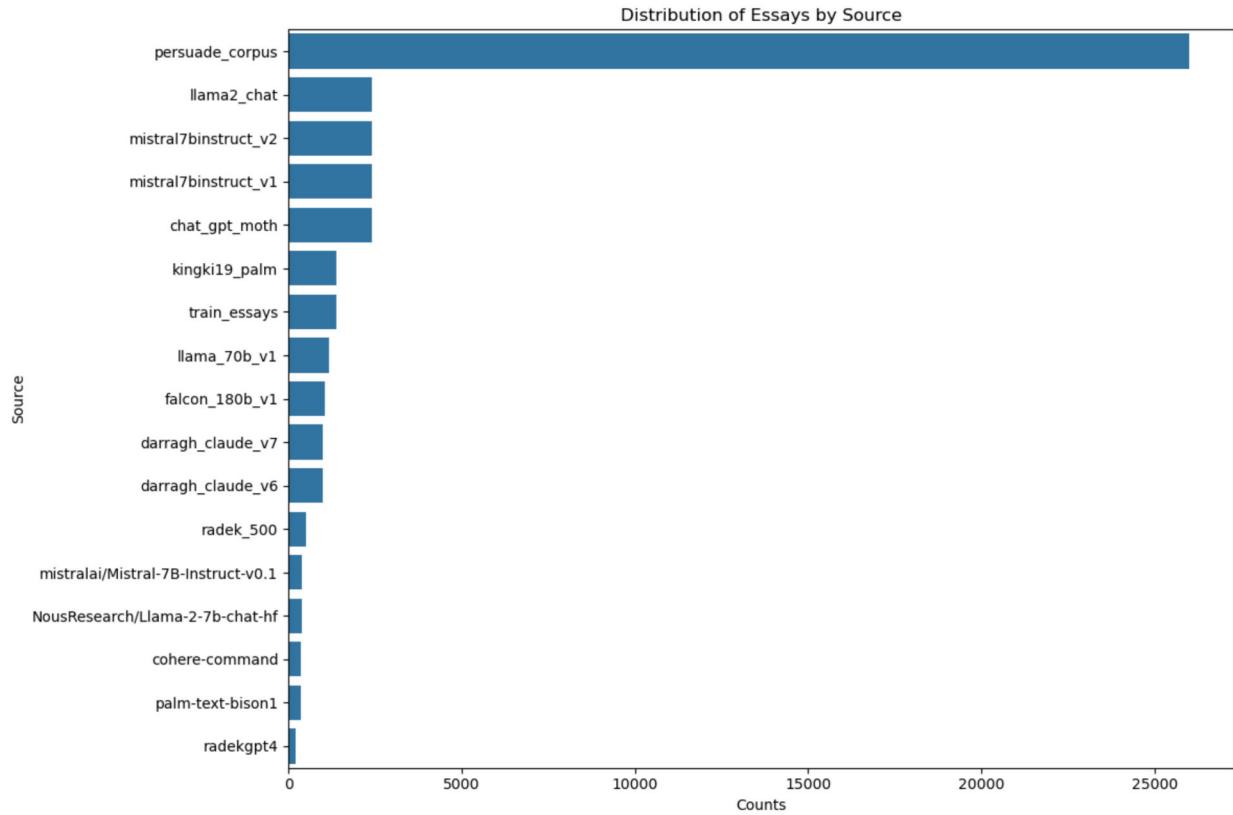


Figure 4. There are over 15 different large language models (LLMs) represented, while the human-written essays are all from the `persuade_corpus`.

III. Data Preparation

Initially, a data cleaning and exploration:

- Missing Values: Identify and address any missing data within the essays or metadata.
- Data Analysis such as Text Length Comparison: Compare the length of AI-generated and human-written essays to identify any initial patterns or discrepancies.
- Filtering out "stopwords", or common English words that contributes little to sentence meaning using the NLTK Python package.
- Feature vector derivation using TF-IDF (Term Frequency-Inverse Document Frequency) for further use.

```

•[15]: #nltk.download('stopwords')
stop_words = set(stopwords.words('english'))

# Stopwords preprocessing
def preprocess_text(text):
    # Remove punctuation
    text = text.translate(str.maketrans("", "", string.punctuation))
    # Remove stopwords
    text = ' '.join([word for word in text.split() if word.lower() not in stop_words])
    return text

# Apply the preprocessing to the essay texts
train_data['text'] = train_data['text'].apply(preprocess_text)
val_data['text'] = val_data['text'].apply(preprocess_text)

[16]: # Vectorize the essays using TF-IDF
vectorizer = TfidfVectorizer(max_features=10000, ngram_range=(1, 2))
X_train = vectorizer.fit_transform(train_data['text'])
X_val = vectorizer.transform(val_data['text'])

# Labels
y_train = train_data['label']
y_val = val_data['label']

•[62]: # Show sample data that have stopwords removed
train_data['text']

[62]: 5233    Luke seems like program help people countries...
28816   One strategy business increasingly adopting co...
1550    Imagine youre populated city Lets say New York...
1888    People tend use cars much basically cant live ...
16987    Driveless cars great enhancement futures socie...
...
11284    believe using technology could effective class...
44732    Dear Senator writing today express support abo...
38158    Despite many hurdles must face attempting achi...
860     Texting Driving Using phone driving worse comb...
15795    Dear Principal really good quarter students wo...
Name: text, Length: 35894, dtype: object

•[64]: # Showing sample vectorized essays
X_train

```

Figure 5. Essays with stopwords removed and vectorized, with training samples shown

IV. Methodology and Results

First, I used a simple logistic regression as a benchmark performance:

```

•[46]: # A simple logistic regression classifier as benchmark
clf = LogisticRegression(max_iter=1000)
clf.fit(X_train, y_train)

# Predict on the validation set
y_pred = clf.predict(X_val)

# Evaluate the classifier
accuracy = accuracy_score(y_val, y_pred)
print("Validation Accuracy: {:.4f}")
print("Classification Report:")
print(classification_report(y_val, y_pred))

Validation Accuracy: 0.6405
Classification Report:
precision    recall  f1-score   support
      0       0.65     0.89     0.75      5481
      1       0.59     0.24     0.35      3493

   accuracy                           0.64      8974
  macro avg       0.62     0.57     0.55      8974
weighted avg       0.63     0.64     0.59      8974

```

Figure 6. Simple Logistic Regression Yields 64% accuracy.

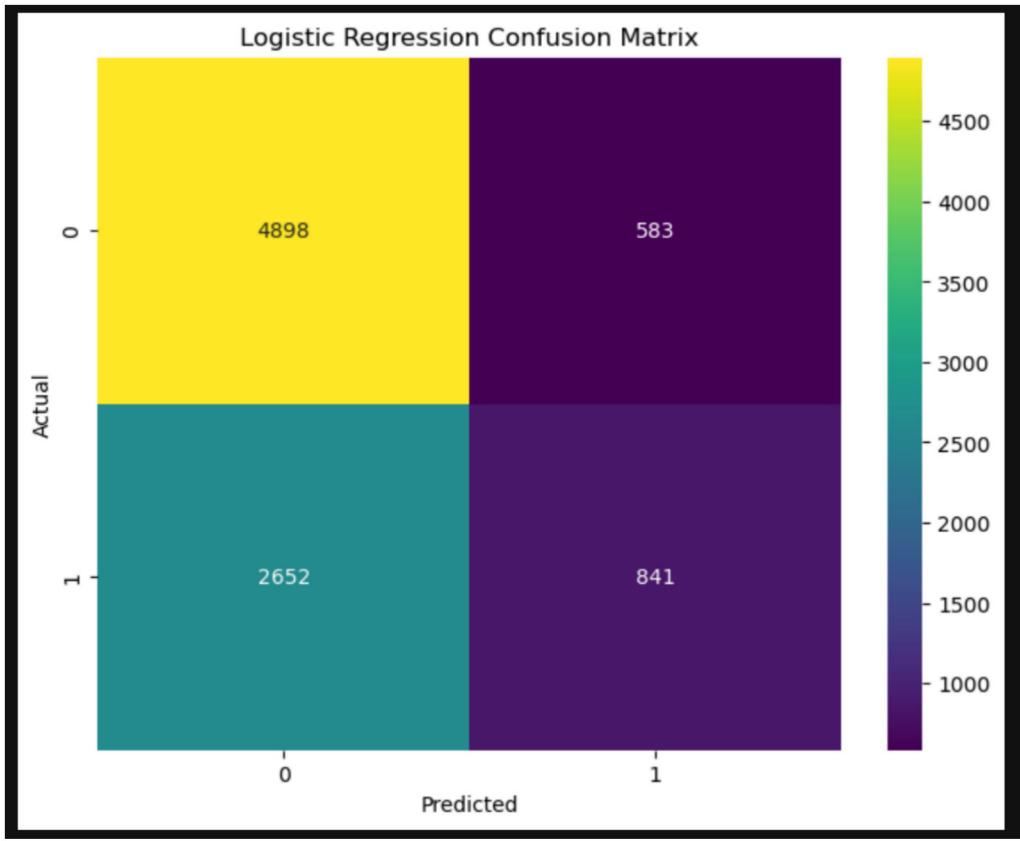


Figure 7. Logistic Regression Confusion Matrix (AI = 1, Human = 0). We see most of the error occurs when AI written text is classified as Human written

Note: After applying better data preprocessing (vectorizing instead of tokenizing) the logistic regression also yields 99% accuracy.

For this text classification task, I chose to use LSTM and RNN as initial models.

LSTM Model Setup and Results

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout

# Build the LSTM model
model = Sequential()
model.add(Embedding(max_features, 128, input_length=maxlen))
model.add(LSTM(64, return_sequences=True))
model.add(Dropout(0.5))
model.add(LSTM(64))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
lstm_history = model.fit(X_train, y_train, epochs=5, batch_size=32, validation_data=(X_val, y_val))

```

Figure 8. The layering and hyperparameterization chosen for the LSTM is seen above.

Layering and hyperparameter choices of the LSTM were as follows:

- Input: Embedding layer with embedding vector dimension set to 128 and max length of 150 set prior
- Two LSTM layers with 64 neuron units, with first layer returning full sequence output to stack layers
- Two dropout layers with 50% dropout to zero each, to prevent overfitting
- A single unit dense layer with sigmoid activation function, since we want a classification output between 0 and 1.
- Adam optimizer to adjust learning rate during training
- Binary crossentropy for classification problem
- Accuracy tracked as the optimized metric

Preliminary results were as follows:

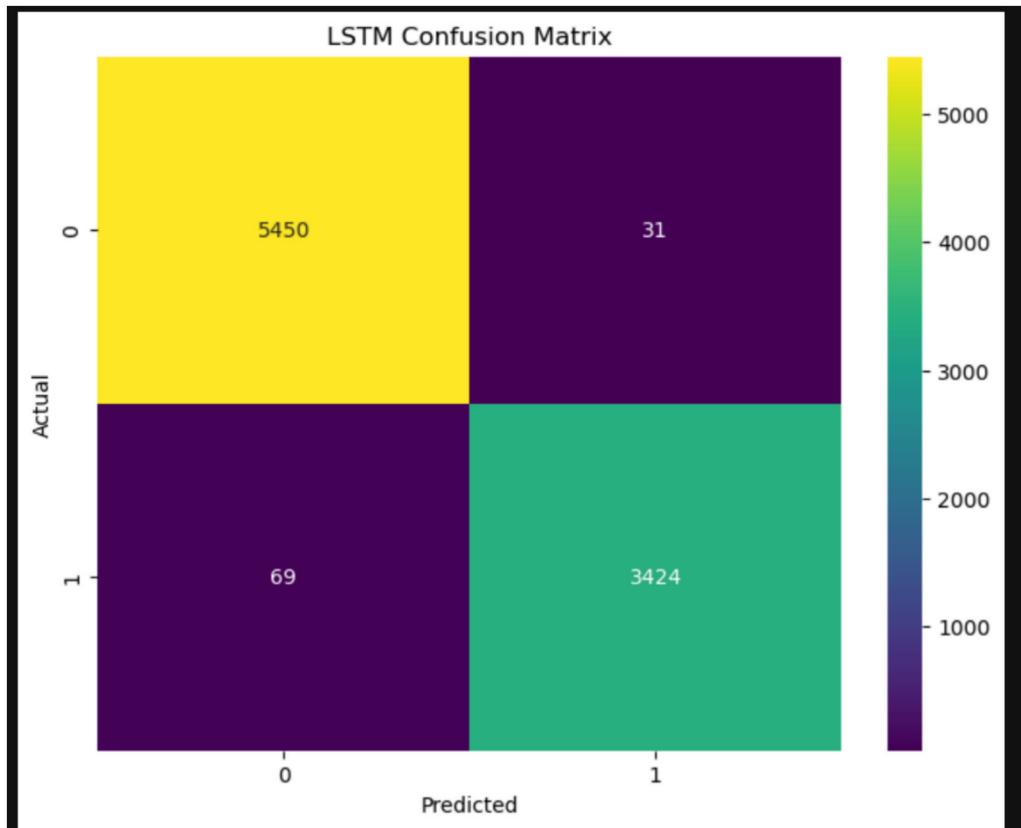


Figure 9. LSTM validation set Confusion Matrix (AI = 1, Human = 0).
Only 100 false positive or false negatives occurred out of 8974 validation essays.

```

LSTM Validation Accuracy: 0.9889
LSTM Classification Report:
precision    recall    f1-score   support
          0       0.99      0.99      0.99      5481
          1       0.99      0.98      0.99      3493

   accuracy                           0.99      8974
macro avg       0.99      0.99      0.99      8974
weighted avg    0.99      0.99      0.99      8974

```

Figure 10. LSTM shows a high 98.89% accuracy on the validation set.

Upon some trial and error, these layering and hyperparameter choices kept the LSTM simple and not too large while still obtaining high accuracy, precision, and recall on the validation set.

RNN Model Setup and Results

```

[47]: # RNN
from tensorflow.keras.layers import SimpleRNN

# Build the RNN model
model_rnn = Sequential()
model_rnn.add(Embedding(max_features, 128, input_length=maxlen))
model_rnn.add(SimpleRNN(64, return_sequences=True))
model_rnn.add(Dropout(0.5))
model_rnn.add(SimpleRNN(64))
model_rnn.add(Dropout(0.5))
model_rnn.add(Dense(1, activation='sigmoid'))

model_rnn.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
rnn_history = model_rnn.fit(X_train, y_train, epochs=5, batch_size=32, validation_data=(X_val, y_val))

Epoch 1/5
1122/1122 [=====] - 31s 27ms/step - loss: 0.4432 - accuracy: 0.7798 - val_loss: 0.5466 - val_accuracy: 0.8015
Epoch 2/5
1122/1122 [=====] - 31s 27ms/step - loss: 0.4498 - accuracy: 0.7941 - val_loss: 0.2333 - val_accuracy: 0.9237
Epoch 3/5
1122/1122 [=====] - 31s 28ms/step - loss: 0.2099 - accuracy: 0.9240 - val_loss: 0.2047 - val_accuracy: 0.9445
Epoch 4/5
1122/1122 [=====] - 31s 28ms/step - loss: 0.0761 - accuracy: 0.9787 - val_loss: 0.0933 - val_accuracy: 0.9748
Epoch 5/5
1122/1122 [=====] - 31s 27ms/step - loss: 0.1110 - accuracy: 0.9706 - val_loss: 0.0691 - val_accuracy: 0.9799

```

Figure 11. The layering and hyperparameterization chosen for the RNN is seen above.

Layering choices of the RNN were as follows and is very similar to the LSTM:

- Input: Embedding layer with embedding vector dimension set to 128 and max length of 150 set prior
- Two simpleRNN layers with 64 neuron units, with first layer returning full sequence output to stack layers
- Two dropout layers with 50% dropout to zero each, to prevent overfitting
- A single unit dense layer with sigmoid activation function, since we want a classification output between 0 and 1.

- Adam optimizer to adjust learning rate during training
- Binary crossentropy for classification problem
- Accuracy tracked as the optimized metric

Preliminary results were as follows:

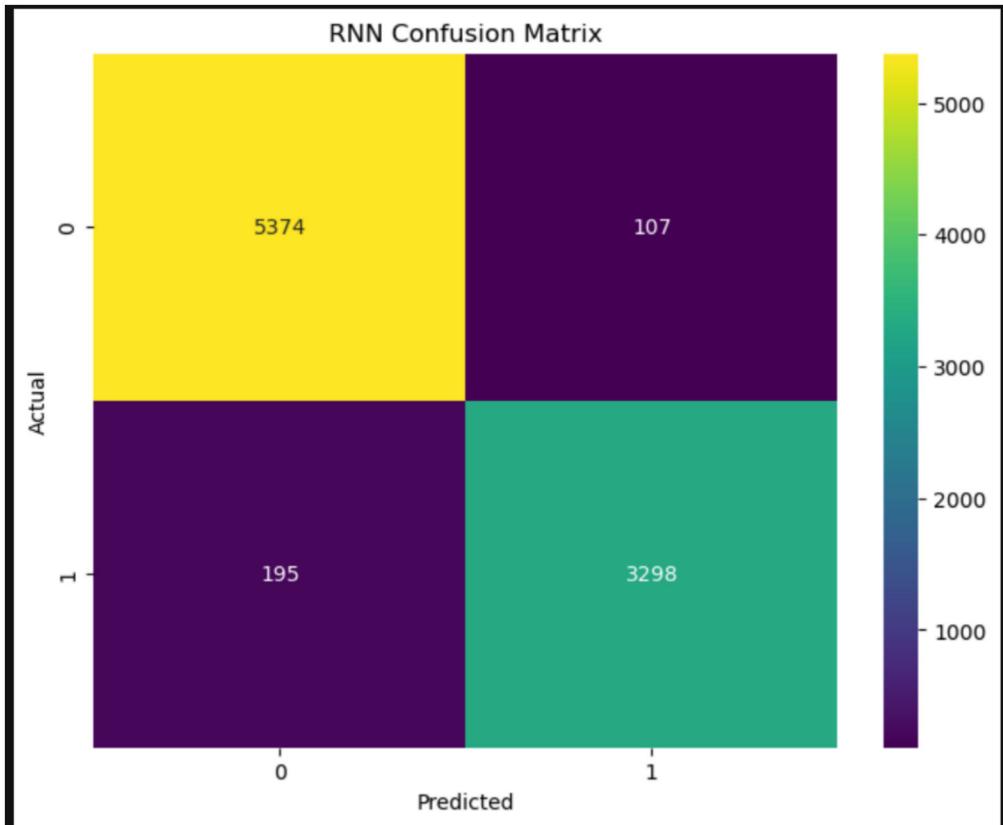


Figure 12. RNN validation set Confusion Matrix (AI = 1, Human = 0).
Only 302 false positive or false negatives occurred out of 8974 validation essays.

```
281/281 [=====] - 2s 6ms/step
RNN Validation Accuracy: 0.9799
RNN Classification Report:
precision    recall   f1-score   support
      0       0.98     0.99     0.98      5481
      1       0.98     0.97     0.97      3493

accuracy                           0.98
macro avg                           0.98
weighted avg                        0.98
```

Figure 13. LSTM shows a high 97.99% accuracy on the validation set.

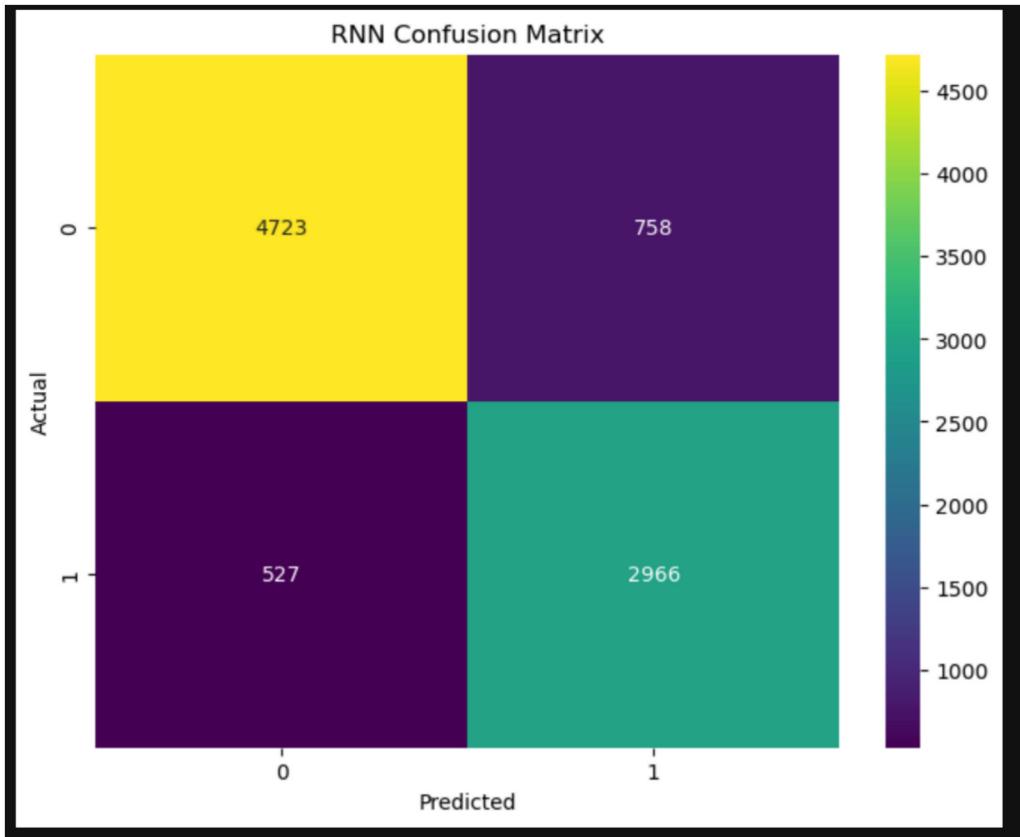


Figure 14. RNN validation set Confusion Matrix (AI = 1, Human = 0).
 Results were much worse when using a different set of layering and parameters.

The RNN hyperparameters were kept similar to the LSTM hyperparameters and not much additional trial and error was done, as it made model performance worse shown in Figure 14.

A Bidirectional LSTM was also tried with the exact same layering and hyperparameters, except I used `Bidirectional(LSTM(64))`, wrapping an LSTM layer to create a bi-directional layer. Results were very similar with a c. 98% accuracy, as the hypothesis is that going through the text in reverse order won't make much difference and the existing accuracy was already very high. The only difference may be robustness.

Discussion on Training

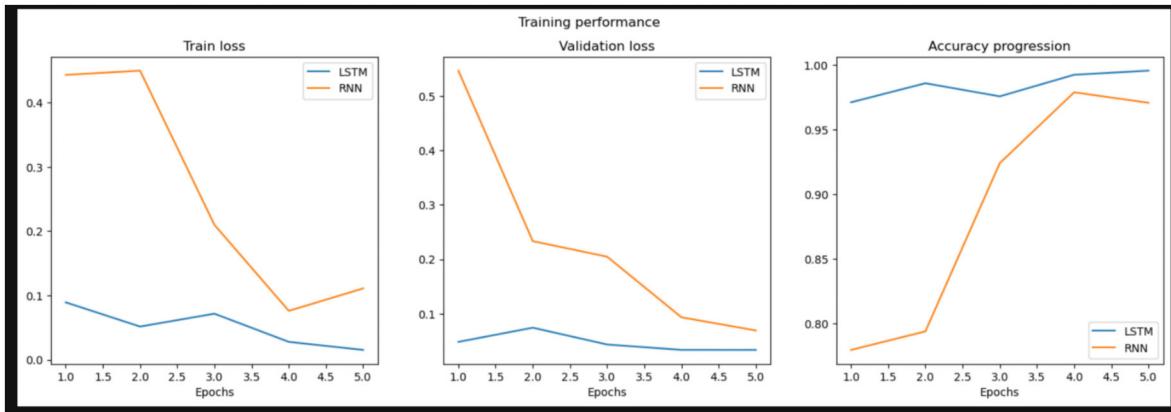


Figure 15. Training and Validation loss for LSTM and RNN over 5 epochs

The LSTM and RNN both training under 10 minutes for 5 epochs, given a batch size of 32. LSTM training time is longer, slightly more than 2x per epoch which is roughly expected given the architecture difference.

LSTM starts off at a very high accuracy while the RNN climbs quickly in around 3 -4 epochs. We can see that the validation loss flatlines earlier for the LSTM as well. Both achieve very high accuracy rates of 98%-99%.

Given the 98% high accuracy of the LSTM right in the first epoch. We fear there may be overfitting and that truly “out of sample” text, i.e. essays that are out of context, or none essay text that is much shorter than the samples, may perform more poorly when the LSTM is used. However note that the scope of this project is not a general AI vs Human text detector given the limited time and resources, and is limited to AI generated vs pre-college student essays.

Breakdown Analysis of Results

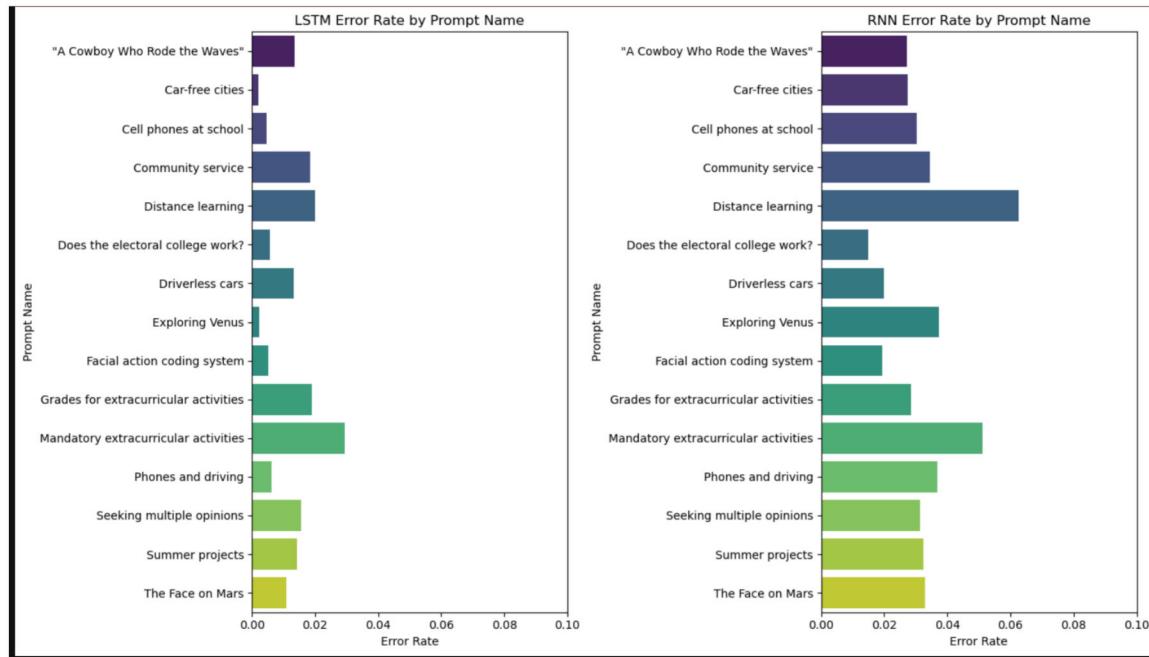


Figure 16. While Error Rates are low, there are clearly prompts that are harder to classify with both LSTM and RNN

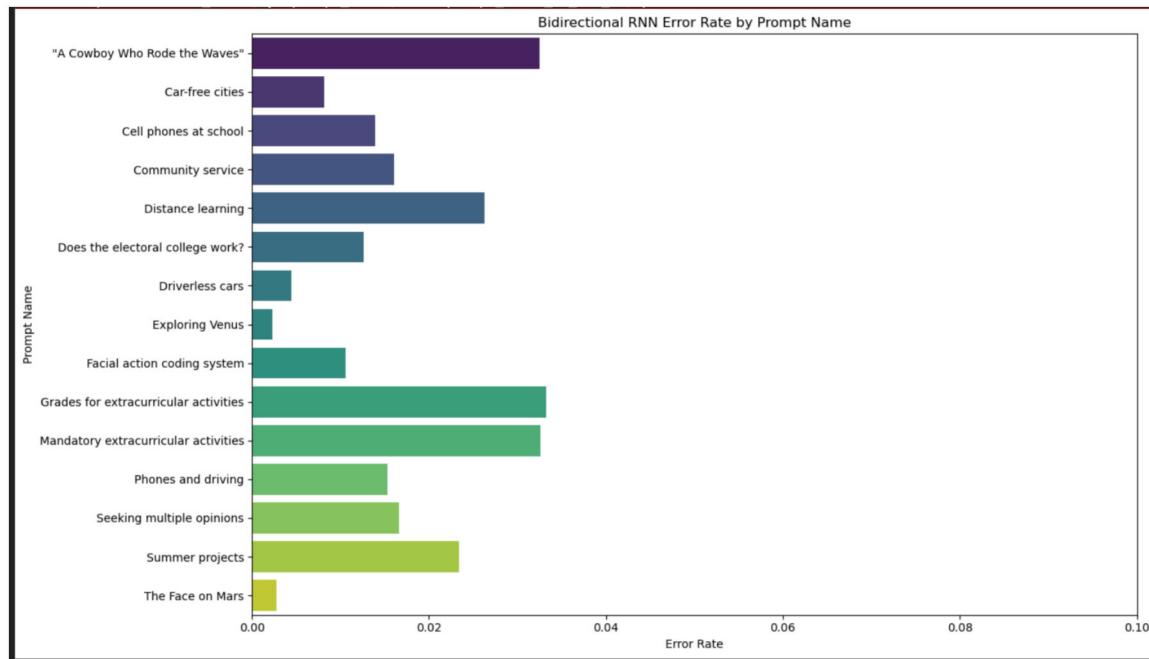


Figure 17. Bidirectional LSTM shows similarly low error rates

V. Simple App Demonstrations and Discussion

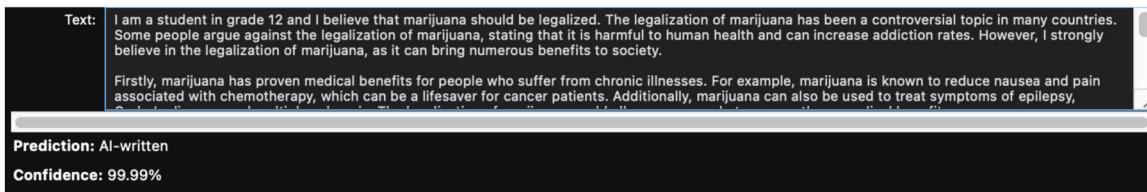


Figure 18. An AI essay copied from the dataset used,
correctly classified with 99.99% confidence using Bidirectional LSTM

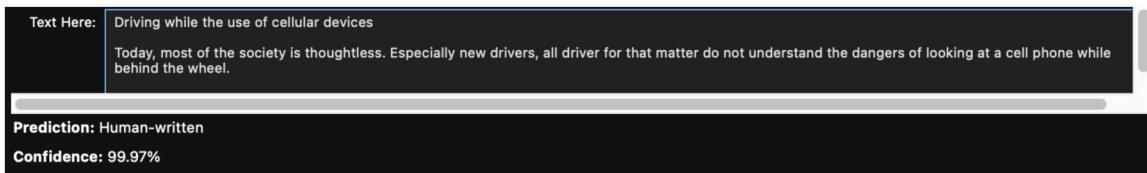


Figure 19. A human written essay copied from the dataset used,
correctly classified with 99.97% confidence using Bidirectional LSTM

I have created a simple widget that can use any saved model to classify a chunk of input text as human or AI written. All tests below will use the bidirectional LSTM trained.

Shown above are two sample Figures 18 and 19, which shows that the model does indeed classify AI and human written essays with c.99% confidence from the DAIGT v2 dataset used.

Next, I used ChatGPT4o by OpenAI, one of the most cutting edge and most popularly used Large Language Models to generate two sample essays, one with the same prompt name as some essays in the training dataset - “Mandatory Extracurricular Activities”, and one prompt name that is out of sample completely - “Fibre Optics in Cloud Computing. The generated essay with the same prompt name as training dataset was correctly classified as AI written as shown in Figure 20. However, the prompt name that is out of sample complete was incorrectly classified as Human written, albeit only with 74% confidence. I suspect this may have to do with the vocabulary being used is completely out of sample, as fibre optics or cloud computing did not show up at all in the training dataset.

Please generate a 200 word essay on "Mandatory Extracurricular Activities"



Mandatory extracurricular activities have become a subject of significant debate in the educational landscape. Proponents argue that these activities are essential for the holistic development of students, providing them with opportunities to acquire skills beyond the academic curriculum. They believe that participation in sports, arts, and various clubs fosters teamwork, leadership, and time management skills, which are crucial for personal growth and future professional success. Additionally, mandatory extracurricular activities can help students discover new interests and talents, promoting a well-rounded education.

On the other hand, critics contend that making extracurricular activities compulsory may place undue pressure on students. They argue that mandatory participation can lead to increased stress and burnout, particularly for those who already struggle with academic demands. Critics also point out that students should have the freedom to choose how they spend their time outside the classroom, emphasizing the importance of personal choice and autonomy. Moreover, mandatory activities may not account for individual differences in interests, capabilities, and home responsibilities, potentially leading to disengagement and resentment.

In conclusion, while mandatory extracurricular activities offer significant benefits in terms of skill development and personal growth, it is crucial to balance these advantages with considerations of student well-being and autonomy. A more flexible approach that encourages, rather than compels, participation may be a more effective way to nurture well-rounded individuals.

Text Here: Mandatory extracurricular activities have become a subject of significant debate in the educational landscape. Proponents argue that these activities are essential for the holistic development of students, providing them with opportunities to acquire skills beyond the academic curriculum. They believe that participation in sports, arts, and various clubs fosters teamwork, leadership, and time management skills, which are crucial for personal growth and future professional success. Additionally, mandatory extracurricular activities can help students discover new interests and talents, promoting a well-rounded education.

Prediction: AI-written

Confidence: 99.95%

Figure 20. A new OpenAI ChatGPT 4o written essay on "Mandatory Extracurricular Activities", a prompt name that is in sample, correctly classified with 99.95% confidence using Bidirectional LSTM

Please generate a 200 word essay on "Fibre Optics in Cloud Computing"



Fibre optics play a pivotal role in the advancement of cloud computing, revolutionizing data transmission and connectivity. Fibre optic technology, which uses light to transmit data at incredibly high speeds, has become the backbone of modern cloud infrastructure. Its capacity to handle large volumes of data with minimal latency is crucial for the efficient functioning of cloud services, which rely on rapid data transfer between data centers and end-users.

One of the primary benefits of fibre optics in cloud computing is its exceptional bandwidth. Fibre optic cables can carry vast amounts of data simultaneously, facilitating seamless access to cloud-based applications and services. This high bandwidth is essential for supporting the growing demand for cloud services, including data storage, processing, and real-time analytics. Additionally, fibre optics offer low latency and high reliability, ensuring that data is transmitted quickly and accurately, which is critical for applications requiring real-time interaction, such as video conferencing, online gaming, and financial transactions.

Furthermore, fibre optics contribute to enhanced security in cloud computing. The technology is less susceptible to electromagnetic interference and eavesdropping, providing a more secure medium for data transmission. As data breaches and cyber threats become more sophisticated, the inherent security advantages of fibre optics are increasingly valuable.



In conclusion, fibre optics are integral to the efficiency, speed, and security of cloud computing.

Text Here: Fibre optics play a pivotal role in the advancement of cloud computing, revolutionizing data transmission and connectivity. Fibre optic technology, which uses light to transmit data at incredibly high speeds, has become the backbone of modern cloud infrastructure. Its capacity to handle large volumes of data with minimal latency is crucial for the efficient functioning of cloud services, which rely on rapid data transfer between data centers and end-users.
One of the primary benefits of fibre optics in cloud computing is its exceptional bandwidth. Fibre optic cables can carry vast amounts of data

Prediction: Human-written
Confidence: 74.08%

Figure 21. A new OpenAI ChatGPT 4o written essay on "Fibre Optics in Cloud Computing", a prompt name that is out of sample, incorrectly classified as human with 74.08% confidence using Bidirectional LSTM

Please generate a 200 word essay on "Phones and Distance Learning"



Phones have emerged as vital tools in the realm of distance learning, offering unprecedented access to educational resources and fostering new modes of communication between students and educators. With the proliferation of smartphones, learners can access course materials, participate in virtual classrooms, and collaborate with peers from virtually anywhere, making education more flexible and accessible.

One of the significant advantages of using phones in distance learning is their portability. Students can carry their entire course load in their pockets, accessing textbooks, lecture notes, and multimedia resources at their convenience. This flexibility is particularly beneficial for those who juggle education with other responsibilities, such as work or family commitments. Additionally, smartphones support various educational apps and platforms that enhance learning experiences through interactive content, quizzes, and real-time feedback.

Communication is another area where phones excel in distance learning. Instant messaging, video calls, and email facilitate continuous interaction between students and instructors, fostering a supportive learning environment. These tools enable prompt feedback and clarification of doubts, which is crucial for maintaining academic progress and motivation.

Text Here: Phones have emerged as vital tools in the realm of distance learning, offering unprecedented access to educational resources and fostering new modes of communication between students and educators. With the proliferation of smartphones, learners can access course materials, participate in virtual classrooms, and collaborate with peers from virtually anywhere, making education more flexible and accessible.

One of the significant advantages of using phones in distance learning is their portability. Students can carry their entire course load in their pockets, accessing textbooks, lecture notes, and multimedia resources at their convenience. This flexibility is particularly beneficial for those who juggle education with other responsibilities, such as work or family commitments. Additionally, smartphones support various educational apps and platforms that enhance learning experiences through interactive content, quizzes, and real-time feedback.

Prediction: AI-written
Confidence: 99.96%

Figure 22. A new OpenAI ChatGPT 4o written essay on "Phones and Distance Learning", a prompt that is a combination of in sample prompts, correctly classified with 99.96% confidence using Bidirectional LSTM

Please generate a 200 word essay on "Distance Learning using Fibre Optics"



Distance learning has been significantly enhanced by the advent of fibre optics technology, which has transformed how education is delivered and accessed. Fibre optics, which uses light to transmit data at incredibly high speeds, has become a cornerstone for modern internet infrastructure, enabling robust and reliable connections essential for distance learning.

One of the key benefits of fibre optics in distance learning is its unparalleled speed and bandwidth. Fibre optic cables can handle vast amounts of data simultaneously, allowing for seamless streaming of high-definition video lectures, real-time virtual classrooms, and rapid downloading of large educational resources. This capability is crucial for providing a smooth and uninterrupted learning experience, especially as online education increasingly incorporates multimedia content and interactive elements.

Furthermore, fibre optics provide low latency and high reliability, which are vital for synchronous learning activities such as live classes and group discussions. Low latency ensures that interactions between students and educators occur in real time without delays, fostering a more engaging and effective learning environment. The high reliability of fibre optic networks also means fewer disruptions, which is essential for maintaining the continuity and consistency of education.



Text Here: Distance learning has been significantly enhanced by the advent of fibre optics technology, which has transformed how education is delivered and accessed. Fibre optics, which uses light to transmit data at incredibly high speeds, has become a cornerstone for modern internet infrastructure, enabling robust and reliable connections essential for distance learning.

One of the key benefits of fibre optics in distance learning is its unparalleled speed and bandwidth. Fibre optic cables can handle vast amounts of data simultaneously, allowing for seamless streaming of high-definition video lectures, real-time virtual classrooms, and rapid downloading of large educational

Prediction: AI-written

Confidence: 99.97%

Figure 23. A new OpenAI ChatGPT 4o written essay on "Distance Learning using Fibre Optics", a prompt that is a combination of in sample and out of sample prompts, correctly classified with 99.97% confidence using Bidirectional LSTM

Finally, given the tests above, I tried two additional prompt names, one that combines in-sample prompt names - “Phones and Distance Learning.” This was correctly identified as AI written with 99.96% accuracy, no surprises given in-sample robustness is expected with deep neural networks in general.

The second new prompt is the combination of an in-sample and out of-sample prompt - “Distance Learning using Fibre Optics.” This was also correctly identified as AI written with 99.97% accuracy, suggesting that the bidirectional LSTM trained uses the existing AI corpus to recognize AI written texts, even if out of sample vocabulary and topics are blended in.

The potential implication here is that with a much larger labeled dataset that inductively exhausts the English human language as well as a much larger model, the classifier would likely be able to identify nearly all AI written essays correctly, assuming no major complications arise in scaling and the “Scaling Law” holds.

In this hypothetically implied scenario, false negatives where AI written essays were classified as human written, as we saw in Figure 21, will drop to near zero. Only false positives will exist if humans mimic AI writing and this will be a much harder problem, if not impossible one to solve.

A final test on robustness to very short texts:

Text Here: Hello my name is Jason, I am a human. You are classifying this incorrectly but the prompt is too short. I am making mistakes on porpoise but you are not picking them up.

Prediction: AI-written
Confidence: 99.69%

Text Here: Hello my name is Jason, I am a human. You are classifying this incorrectly but the prompt is too short. I am making mistakes on porpoise but you are not picking them up.

Future think people dont think good idea think people dont want people think people dont want people dont

Prediction: Human-written
Confidence: 99.55%

Figure 24. The classifier is not robust to very short texts as shown above with the statement I wrote, even if I try to make an obvious spelling mistake. However, given clear indication of say repetition via copy and pasting, something an AI large language model will never output, the classifier then correctly classifies the text as human written.

VI. Conclusion

A DAIGTv2 dataset using more than 15,000 AI-written essays and more than 25,000 pre-college student human written essays were used for training and validation. An LSTM, simple RNN, and Bidirectional LSTM was trained for the classification task, with dropout layers included to prevent overfitting. These models achieved a 97-99% accuracy on validation data, compared to a benchmark simple logistic regression that only achieved a 64% accuracy.

A simple application was also created with a textbox that can be used to input text, which then shows a classification of whether the text was written by AI or human and a confidence percentage associated with the classification. Individual in-sample and out-of-sample prompts were tested for analysis.

There are some obvious limitations in essays AI-generated by LLMs. For example, any essays with incorrect mathematical calculations are highly likely to be AI-generated, since LLMs currently struggle with arithmetic. Currently mainstream LLMs also output with an upper limit in text length and have limited context windows in terms of interpreting input.

With our observations, a preliminary conclusion was reached that the “Scaling Law” of increased data resulting in increased accuracy and robustness should hold for this use case given more data, specifically to decrease false negative rates (AI-written incorrectly classified as human written). The fundamental hypothesis is that zero shot AI-written essays are done without the AI ever pressing the backspace key or making edits and so the hypothesis is that they should exhibit certain unique characteristics. Our observations support this.

We have also found that synthetic data may be useful for scaling the model and for validation in any further proposed exploration.

All code and the simple application can be found in the Jupyter notebooks/appendix. The full dataset used is found here:

<https://www.kaggle.com/datasets/thedrcat/daigt-v2-train-dataset>

Link to Youtube video is found here:

<https://www.youtube.com/watch?v=FDVWbsIDchk>

```
In [117...]: # Appendix I - All code used
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

import nltk
from nltk.corpus import stopwords
import string
```

```
In [118...]: # Load the dataset
```

```
# Can be downloaded at https://www.kaggle.com/datasets/thedrcat/daigt-v2-train-and-test-datasets
data = pd.read_csv('train_v2_drcat_02.csv')
```

```
In [119...]: # Display structure of the dataset
```

```
print("Dataset Info:")
print(data.info())

# Check for missing values
print("\nMissing Values:")
print(data.isnull().sum())
```

Dataset Info:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44868 entries, 0 to 44867
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   text              44868 non-null    object 
 1   label             44868 non-null    int64  
 2   prompt_name       44868 non-null    object 
 3   source            44868 non-null    object 
 4   RDizzl3_seven    44868 non-null    bool   
dtypes: bool(1), int64(1), object(3)
memory usage: 1.4+ MB
None
```

Missing Values:

```
text          0
label         0
prompt_name  0
source        0
RDizzl3_seven 0
dtype: int64
```

```
In [120...]: # Examine the data a bit further
print("\nLabel Distribution:")
print(data['label'].value_counts())

# Separate the data into essays generated by AI (label=1) and written by humans
human_written = data[data['label'] == 0]
ai_generated = data[data['label'] == 1]

# Display the first few rows of human-written essays
print("\nHuman-written essays:")
print(human_written.head())

# Display the first few rows of AI-generated essays
print("\nAI-generated essays:")
print(ai_generated.head())
```

Label Distribution:

```
label
0    27371
1    17497
Name: count, dtype: int64
```

Human-written essays:

		text	label	\
0	Phones\n\nModern humans today are always on th...		0	
1	This essay will explain if drivers should or s...		0	
2	Driving while the use of cellular devices\n\nT...		0	
3	Phones & Driving\n\nDrivers should not be able...		0	
4	Cell Phone Operation While Driving\n\nThe abil...		0	

	prompt_name	source	RDizzl3_seven
0	Phones and driving	persuade_corpus	False
1	Phones and driving	persuade_corpus	False
2	Phones and driving	persuade_corpus	False
3	Phones and driving	persuade_corpus	False
4	Phones and driving	persuade_corpus	False

AI-generated essays:

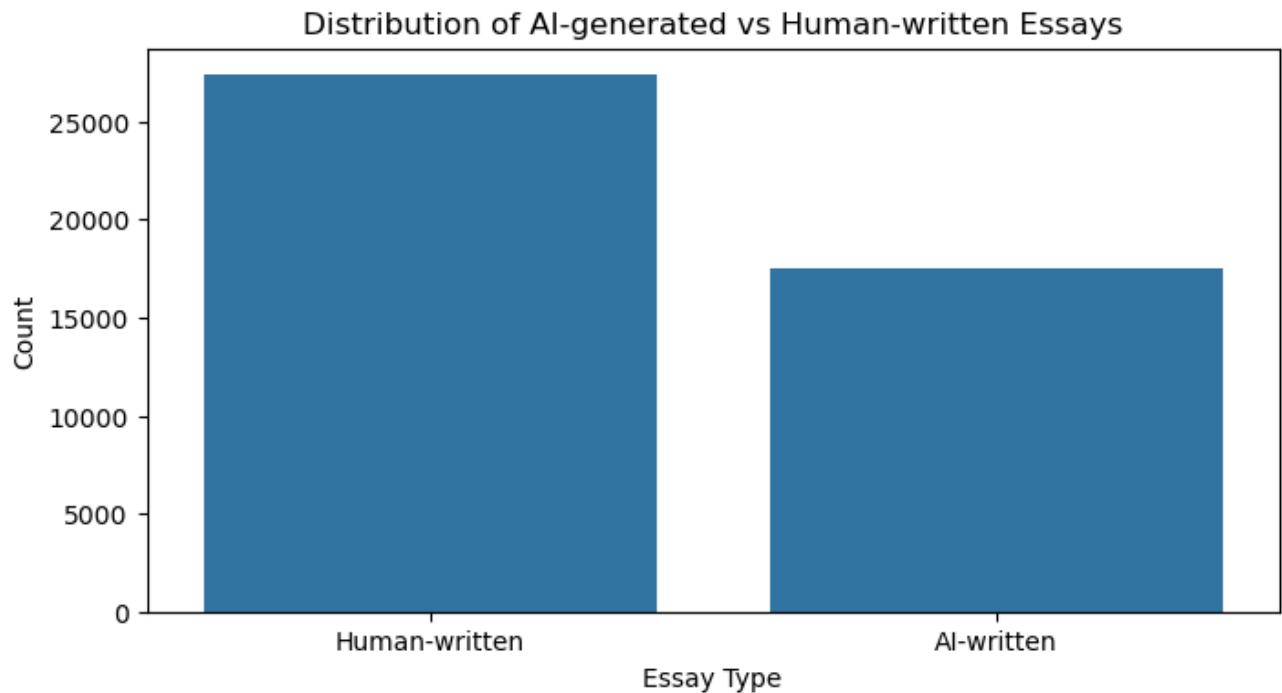
		text	label	\
25996	In recent years, technology has had a profoun...		1	
25997	I strongly believe that meditation and mindful...		1	
25998	One way school administrators can attempt to c...		1	
25999	While summer is meant as a break from the regu...		1	
26000	The use of Facial Action Coding System (FACS) ...		1	

	prompt_name	source	RDizzl3_seven
25996	Car-free cities	mistral7binstruct_v2	True
25997	Distance learning	llama_70b_v1	False
25998	Cell phones at school	chat_gpt_moth	False
25999	Summer projects	darragh_claude_v7	False
26000	Facial action coding system	darragh_claude_v6	True

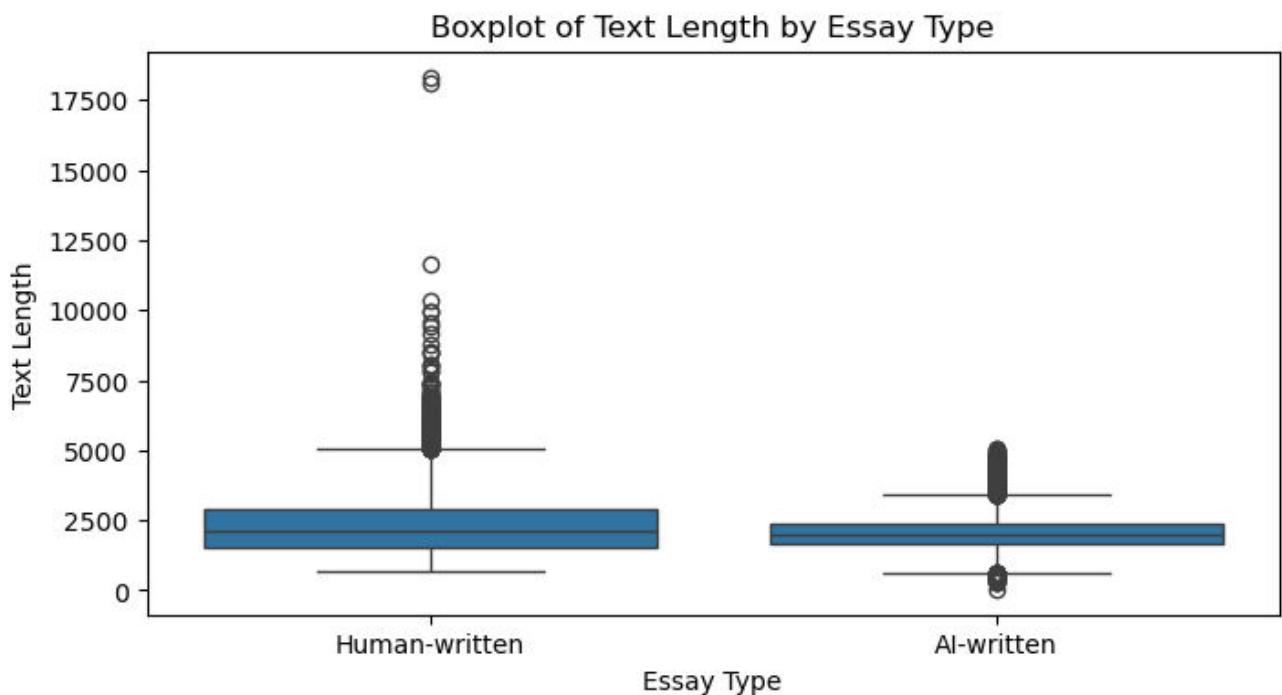
In [121]: # Plot the number of AI-generated vs human-written essays

```
plt.figure(figsize=(8, 4))
sns.countplot(x='label', data=data)
plt.title('Distribution of AI-generated vs Human-written Essays')
plt.xlabel('Essay Type')
plt.ylabel('Count')
plt.xticks(ticks=[0, 1], labels=['Human-written', 'AI-written'])

plt.show()
```



```
In [122]: # Additional analysis: length of essays boxplot
data['text_length'] = data['text'].apply(len)
plt.figure(figsize=(8, 4))
sns.boxplot(x='label', y='text_length', data=data)
plt.title('Boxplot of Text Length by Essay Type')
plt.xlabel('Essay Type')
plt.ylabel('Text Length')
plt.xticks(ticks=[0, 1], labels=['Human-written', 'AI-written'])
plt.show()
```

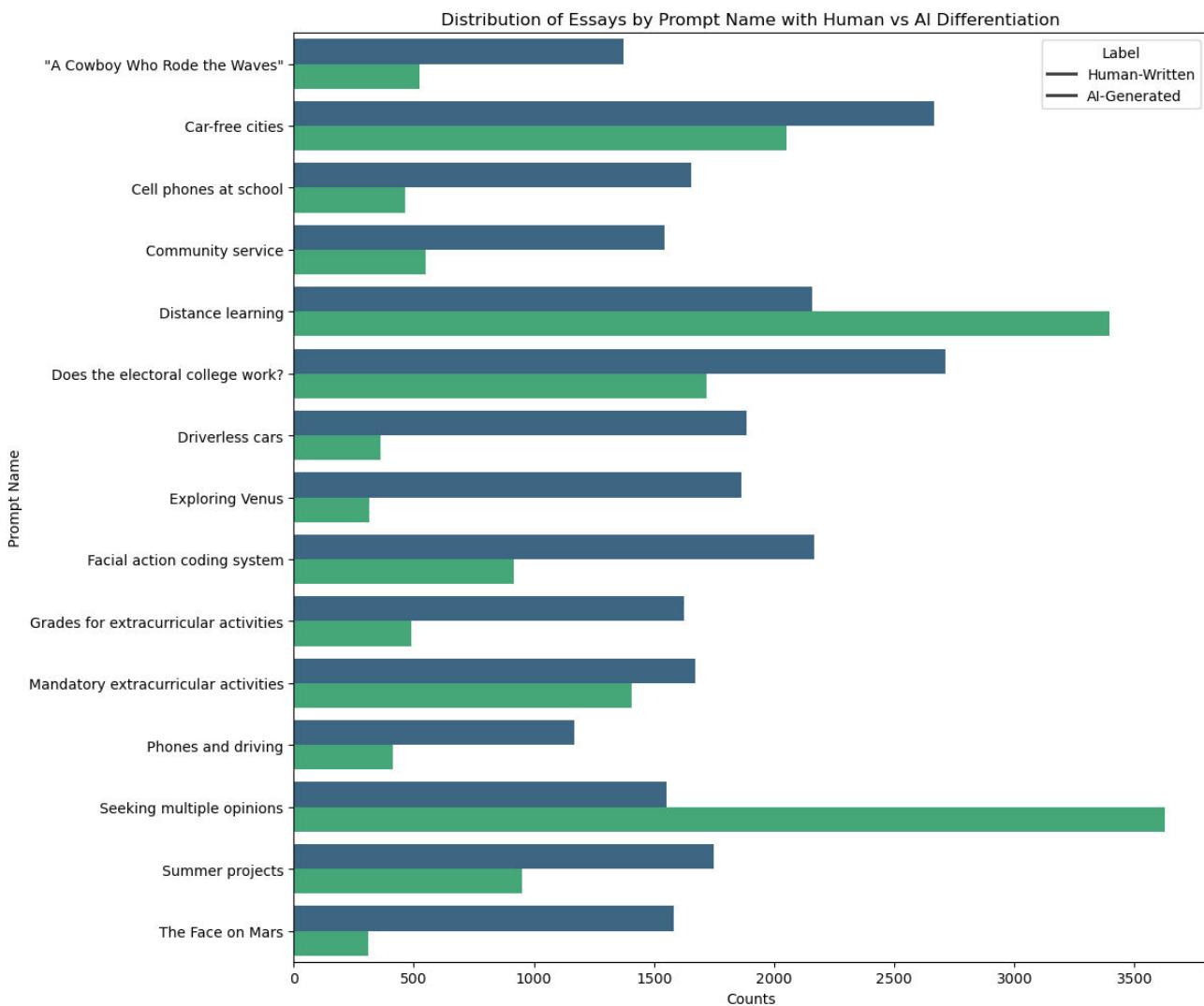


In [123...]

```
# Group data by 'prompt_name' and 'label', and count occurrences
prompt_label_group = data.groupby(['prompt_name', 'label']).size().reset_index()

# Sorting the data by prompt name and counts to better visualize in the plot
prompt_label_sorted = prompt_label_group.sort_values(['prompt_name', 'counts'], ascending=False)

# Plotting the distribution of essays by prompt name, with differentiation by label
plt.figure(figsize=(12, 10))
sns.barplot(x='counts', y='prompt_name', hue='label', data=prompt_label_sorted)
plt.title('Distribution of Essays by Prompt Name with Human vs AI Differentiation')
plt.xlabel('Counts')
plt.ylabel('Prompt Name')
plt.legend(title='Label', labels=['Human-Written', 'AI-Generated'])
plt.tight_layout()
plt.show()
```

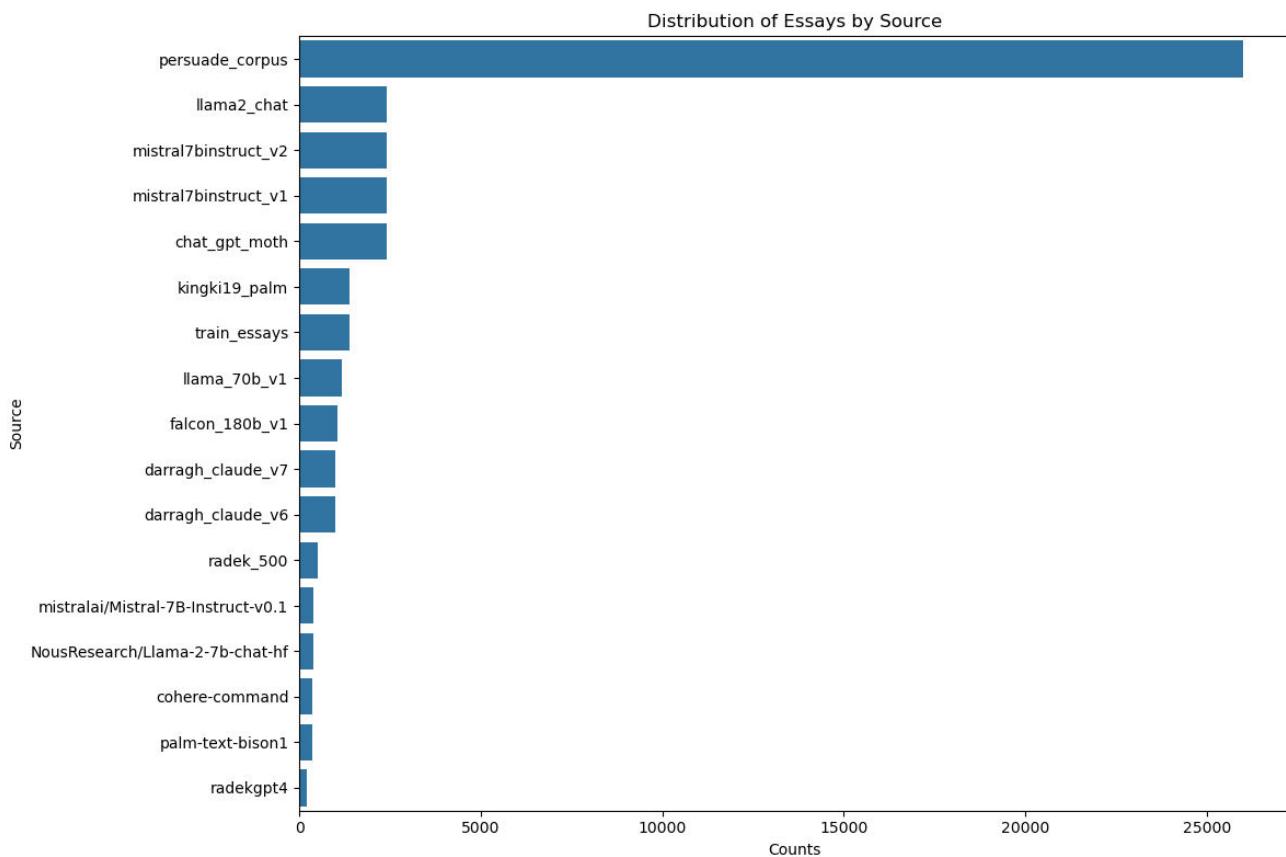


In [124...]

```
# Group data by 'source' and count occurrences
source_group = data.groupby('source').size().reset_index(name='counts')
```

```
# Sort the data by counts before plotting
source_sorted = source_group.sort_values('counts', ascending=False)

# Plotting the distribution of essays by source
plt.figure(figsize=(12, 8))
sns.barplot(x='counts', y='source', data=source_sorted)
plt.title('Distribution of Essays by Source')
plt.xlabel('Counts')
plt.ylabel('Source')
plt.tight_layout() # Adjusts plot to ensure everything fits without overlap
plt.show()
```



```
In [125]: # Ensure 'label' column is in the dataset
assert 'label' in data.columns, "The dataset does not contain a 'label' column"

# Split the data into training and validation sets
train_data, val_data = train_test_split(data, test_size=0.2, random_state=42)

# Display the shape of the training and validation sets
print(f'Training set shape: {train_data.shape}')
print(f'Validation set shape: {val_data.shape}')
```

Training set shape: (35894, 6)
Validation set shape: (8974, 6)

```
In [126... print(data.columns)
      data.head()
```

```
Index(['text', 'label', 'prompt_name', 'source', 'RDizzl3_seven',
       'text_length'],
      dtype='object')
```

Out[126...]

	text	label	prompt_name	source	RDizzl3_seven	text_length
0	Phones\n\nModern humans today are always on th...	0	Phones and driving	persuade_corpus	False	202
1	This essay will explain if drivers should or s...	0	Phones and driving	persuade_corpus	False	214
2	Driving while the use of cellular devices\n\nT...	0	Phones and driving	persuade_corpus	False	104
3	Phones & Driving\n\nDrivers should not be able...	0	Phones and driving	persuade_corpus	False	124
4	Cell Phone Operation While Driving\n\nThe abil...	0	Phones and driving	persuade_corpus	False	191

In [127...]

```
#nltk.download('stopwords')
stop_words = set(stopwords.words('english'))

# Stopwords preprocessing
def preprocess_text(text):
    # Remove punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))
    # Remove stopwords
    text = ' '.join([word for word in text.split() if word.lower() not in stop_words])
    return text

# Apply the preprocessing to the essay texts
train_data['text'] = train_data['text'].apply(preprocess_text)
val_data['text'] = val_data['text'].apply(preprocess_text)
```

In [128...]

```
# Vectorize the essays using TF-IDF, added to improve logistic regression performance
vectorizer = TfidfVectorizer(max_features=10000, ngram_range=(1, 2))
X_train = vectorizer.fit_transform(train_data['text'])
X_val = vectorizer.transform(val_data['text'])
```

```
# Labels
y_train = train_data['label']
y_val = val_data['label']
```

```
In [129...]: # Show sample data that have stopwords removed
train_data['text']
```

```
Out[129...]: 5233    Luke seems like program help people countries...
28816   One strategy business increasingly adopting co...
1550    Imagine youre populated city Lets say New York...
1888    People tend use cars much basically cant live ...
16987   Driveless cars great enhancement futures socie...
...
11284    believe using technology could effective class...
44732    Dear Senator writing today express support abo...
38158    Despite many hurdles must face attempting achi...
860     Texting Driving Using phone driving worse comb...
15795    Dear Principal really good quarter students wo...
Name: text, Length: 35894, dtype: object
```

```
In [140...]: # Note below shows TOKENIZED and not VECTORIZED X-train (which will be sparse)
X_train
```

```
Out[140...]: array([[ 0,  0,  0, ...,  7, 513,  66],
       [ 0,  0,  0, ..., 606, 3048, 823],
       [5293, 258, 948, ..., 290, 223, 1697],
       ...,
       [ 0,  0,  0, ..., 214, 137,  20],
       [167, 433, 128, ..., 56, 2511,  10],
       [ 0,  0,  0, ..., 466, 135, 273]], dtype=int32)
```

```
In [131...]: # Preprocessing and data exploration complete. Below is model related code
```

```
In [141...]: # A simple logistic regression classifier as benchmark
clf = LogisticRegression(max_iter=1000)
clf.fit(X_train, y_train)

# Predict on the validation set
y_pred = clf.predict(X_val)

# Evaluate the classifier
accuracy = accuracy_score(y_val, y_pred)
print(f'Validation Accuracy: {accuracy:.4f}')
print('Classification Report:')
print(classification_report(y_val, y_pred))
```

Validation Accuracy: 0.6405

Classification Report:

	precision	recall	f1-score	support
0	0.65	0.89	0.75	5481
1	0.59	0.24	0.35	3493
accuracy			0.64	8974
macro avg	0.62	0.57	0.55	8974
weighted avg	0.63	0.64	0.59	8974

In [137]:

```
# Setting up LSTM Model
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

# Parameters
max_features = 10000
maxlen = 150

# Tokenize the text for LSTM
tokenizer = Tokenizer(num_words=max_features)
tokenizer.fit_on_texts(train_data['text'])

X_train = tokenizer.texts_to_sequences(train_data['text'])
X_val = tokenizer.texts_to_sequences(val_data['text'])

# Pad the sequences
X_train = pad_sequences(X_train, maxlen=maxlen)
X_val = pad_sequences(X_val, maxlen=maxlen)
```

In [87]:

```
# Building and Training the LSTM model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout

model = Sequential()
model.add(Embedding(max_features, 128, input_length=maxlen))
model.add(LSTM(64, return_sequences=True))
model.add(Dropout(0.5))
model.add(LSTM(64))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
lstm_history = model.fit(X_train, y_train, epochs=5, batch_size=32, validation_data=(X_val, y_val))
```

Epoch 1/5

```
2024-08-07 01:03:09.116043: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]  
2024-08-07 01:03:09.116487: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_grad(concat/split/split_d  
im' with dtype int32  
    [[{{node gradients/split_grad(concat/split/split_dim)}}]]  
2024-08-07 01:03:09.116987: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]  
2024-08-07 01:03:09.168917: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]  
2024-08-07 01:03:09.169293: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_grad(concat/split/split_d  
im' with dtype int32  
    [[{{node gradients/split_grad(concat/split/split_dim)}}]]  
2024-08-07 01:03:09.169658: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]  
2024-08-07 01:03:09.242601: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]  
2024-08-07 01:03:09.243012: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_grad(concat/split/split_d  
im' with dtype int32  
    [[{{node gradients/split_grad(concat/split/split_dim)}}]]  
2024-08-07 01:03:09.243484: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic
```

```
ate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 01:03:09.294071: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 01:03:09.294485: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 01:03:09.294854: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 01:03:09.507556: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 01:03:09.508003: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 01:03:09.508488: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 01:03:09.557534: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 01:03:09.557943: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim'
```

```
im' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 01:03:09.558368: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim)' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
1122/1122 [=====] - ETA: 0s - loss: 0.0783 - accuracy: 0.9723
2024-08-07 01:04:24.304839: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim)' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 01:04:24.305441: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim)' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 01:04:24.305776: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim)' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 01:04:24.356294: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim)' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 01:04:24.356623: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim)' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 01:04:24.356972: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim)' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
```

```
1122/1122 [=====] - 80s 71ms/step - loss: 0.0783 -  
accuracy: 0.9723 - val_loss: 0.0312 - val_accuracy: 0.9910  
Epoch 2/5  
1122/1122 [=====] - 81s 72ms/step - loss: 0.0239 -  
accuracy: 0.9935 - val_loss: 0.0340 - val_accuracy: 0.9910  
Epoch 3/5  
1122/1122 [=====] - 82s 73ms/step - loss: 0.0197 -  
accuracy: 0.9946 - val_loss: 0.0372 - val_accuracy: 0.9887  
Epoch 4/5  
1122/1122 [=====] - 83s 74ms/step - loss: 0.0118 -  
accuracy: 0.9970 - val_loss: 0.0435 - val_accuracy: 0.9890  
Epoch 5/5  
1122/1122 [=====] - 83s 74ms/step - loss: 0.0084 -  
accuracy: 0.9978 - val_loss: 0.0541 - val_accuracy: 0.9876
```

```
In [88]: # Save the LSTM model  
model.save('lstm_model.h5')
```

```
In [89]: # Predict on the validation set  
y_pred_lstm = model.predict(X_val)  
y_pred_lstm = (y_pred_lstm > 0.5).astype("int32")  
  
# Evaluate the classifier  
accuracy_lstm = accuracy_score(y_val, y_pred_lstm)  
print(f'LSTM Validation Accuracy: {accuracy_lstm:.4f}')  
print('LSTM Classification Report: ')  
print(classification_report(y_val, y_pred_lstm))
```

```
4/281 [.....] - ETA: 4s
```

```
2024-08-07 01:09:58.033339: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 01:09:58.033803: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 01:09:58.034376: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 01:09:58.085657: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 01:09:58.086053: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 01:09:58.086408: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
281/281 [=====] - 5s 18ms/step
LSTM Validation Accuracy: 0.9876
LSTM Classification Report:
      precision    recall   f1-score   support
0         0.99     0.99     0.99      5481
1         0.98     0.99     0.98      3493
accuracy                           0.99      8974
macro avg       0.99     0.99     0.99      8974
weighted avg    0.99     0.99     0.99      8974
```

```
In [ ]: # Note that the Validation accuracies may not be exactly the same as in repre
```

```
In [90]: # Building and Training the RNN model
from tensorflow.keras.layers import SimpleRNN

model_rnn = Sequential()
model_rnn.add(Embedding(max_features, 128, input_length=maxlen))
model_rnn.add(SimpleRNN(64, return_sequences=True))
model_rnn.add(Dropout(0.5))
model_rnn.add(SimpleRNN(64))
model_rnn.add(Dropout(0.5))
model_rnn.add(Dense(1, activation='sigmoid'))

model_rnn.compile(optimizer='adam', loss='binary_crossentropy', metrics=['ac

# Train the model
rnn_history = model_rnn.fit(X_train, y_train, epochs=5, batch_size=32, vali
```

```
Epoch 1/5
1122/1122 [=====] - 31s 27ms/step - loss: 0.2073 - accuracy: 0.9185 - val_loss: 0.0750 - val_accuracy: 0.9786
Epoch 2/5
1122/1122 [=====] - 31s 27ms/step - loss: 0.1802 - accuracy: 0.9220 - val_loss: 0.6251 - val_accuracy: 0.6917
Epoch 3/5
1122/1122 [=====] - 31s 27ms/step - loss: 0.4263 - accuracy: 0.8068 - val_loss: 0.3852 - val_accuracy: 0.8729
Epoch 4/5
1122/1122 [=====] - 31s 27ms/step - loss: 0.2789 - accuracy: 0.8930 - val_loss: 0.1506 - val_accuracy: 0.9595
Epoch 5/5
1122/1122 [=====] - 31s 27ms/step - loss: 0.1387 - accuracy: 0.9579 - val_loss: 0.1192 - val_accuracy: 0.9663
```

```
In [91]: # Predict on the validation set
y_pred_rnn = model_rnn.predict(X_val)
y_pred_rnn = (y_pred_rnn > 0.5).astype("int32")

# Evaluate the classifier
accuracy_rnn = accuracy_score(y_val, y_pred_rnn)
print(f'RNN Validation Accuracy: {accuracy_rnn:.4f}')
print('RNN Classification Report:')
print(classification_report(y_val, y_pred_rnn))
```

```
281/281 [=====] - 2s 6ms/step
RNN Validation Accuracy: 0.9663
RNN Classification Report:
precision    recall    f1-score   support
          0       0.96      0.98      0.97     5481
          1       0.97      0.94      0.96     3493

accuracy                           0.97     8974
macro avg                           0.97      0.96      0.96     8974
weighted avg                        0.97      0.97      0.97     8974
```

```
In [92]: # Save the rnn model
model_rnn.save('rnn_model.h5')
```

```
In [93]: # Extract LSTM and RNN training history
lstm_history = {
    'loss': lstm_history.history['loss'],
    'val_loss': lstm_history.history['val_loss'],
    'accuracy': lstm_history.history['accuracy'],
    'val_accuracy': lstm_history.history['val_accuracy']
}

rnn_history = {
    'loss': rnn_history.history['loss'],
    'val_loss': rnn_history.history['val_loss'],
    'accuracy': rnn_history.history['accuracy'],
    'val_accuracy': rnn_history.history['val_accuracy']
}
```

```
In [94]: # Function to plot both histories on a single plots
def plot_training_history(histories, titles, steps):
    fig, axs = plt.subplots(1, 3, figsize=(18, 5))

    for history, title, step in zip(histories, titles, steps):
        epochs = range(1, len(history['loss']) + 1)
        axs[0].plot(epochs, history['loss'], label=title)
        axs[1].plot(epochs, history['val_loss'], label=title)
        axs[2].plot(epochs, history['accuracy'], label=title)

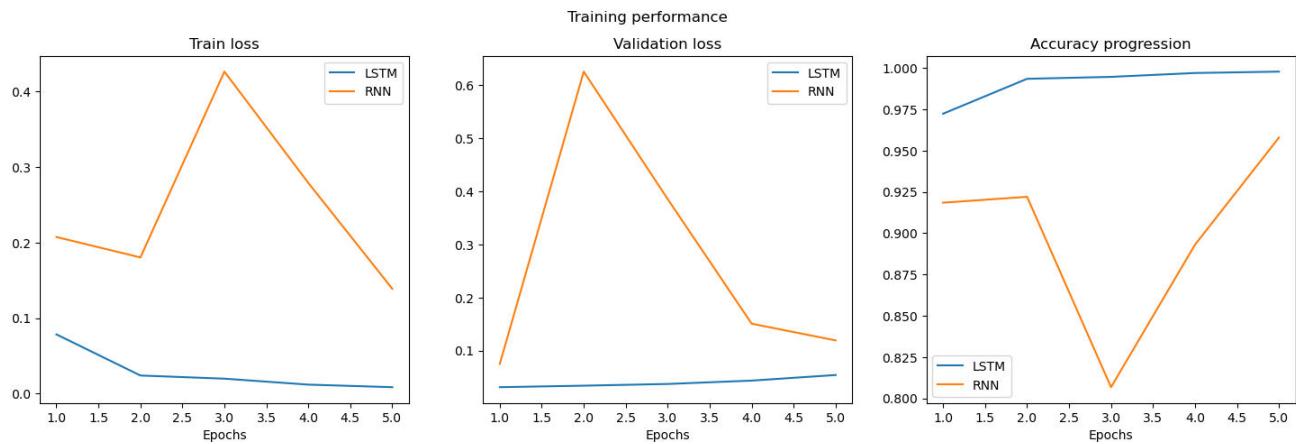
    axs[0].set_title('Train loss')
    axs[1].set_title('Validation loss')
    axs[2].set_title('Accuracy progression')
    for ax in axs:
        ax.legend()
        ax.set_xlabel('Epochs')

    plt.suptitle('Training performance')
```

```
plt.show()

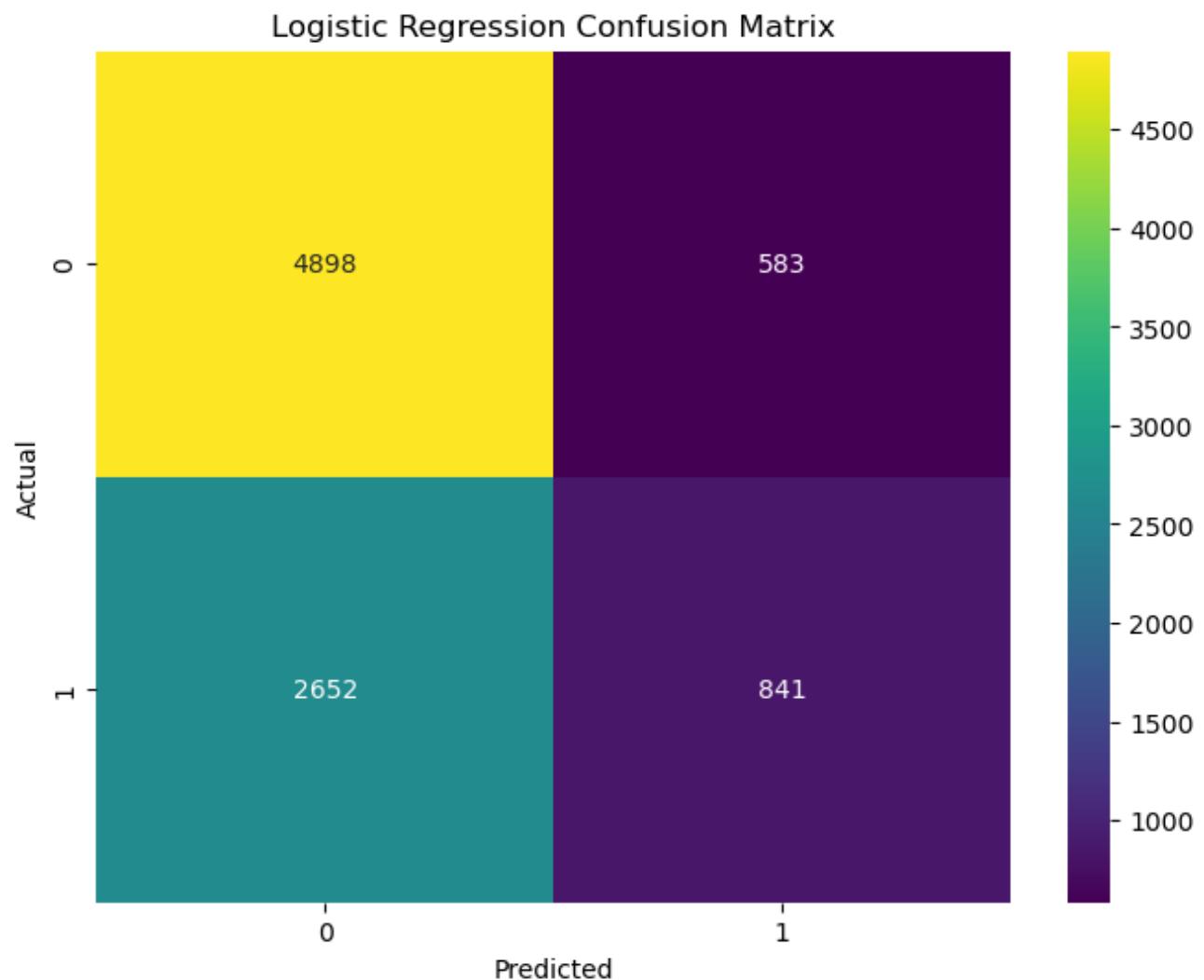
histories = [lstm_history, rnn_history]
titles = ['LSTM', 'RNN']
steps = [step, 1, 1] # steps used for plotting epochs

# Plot the histories
plot_training_history(histories, titles, steps)
```

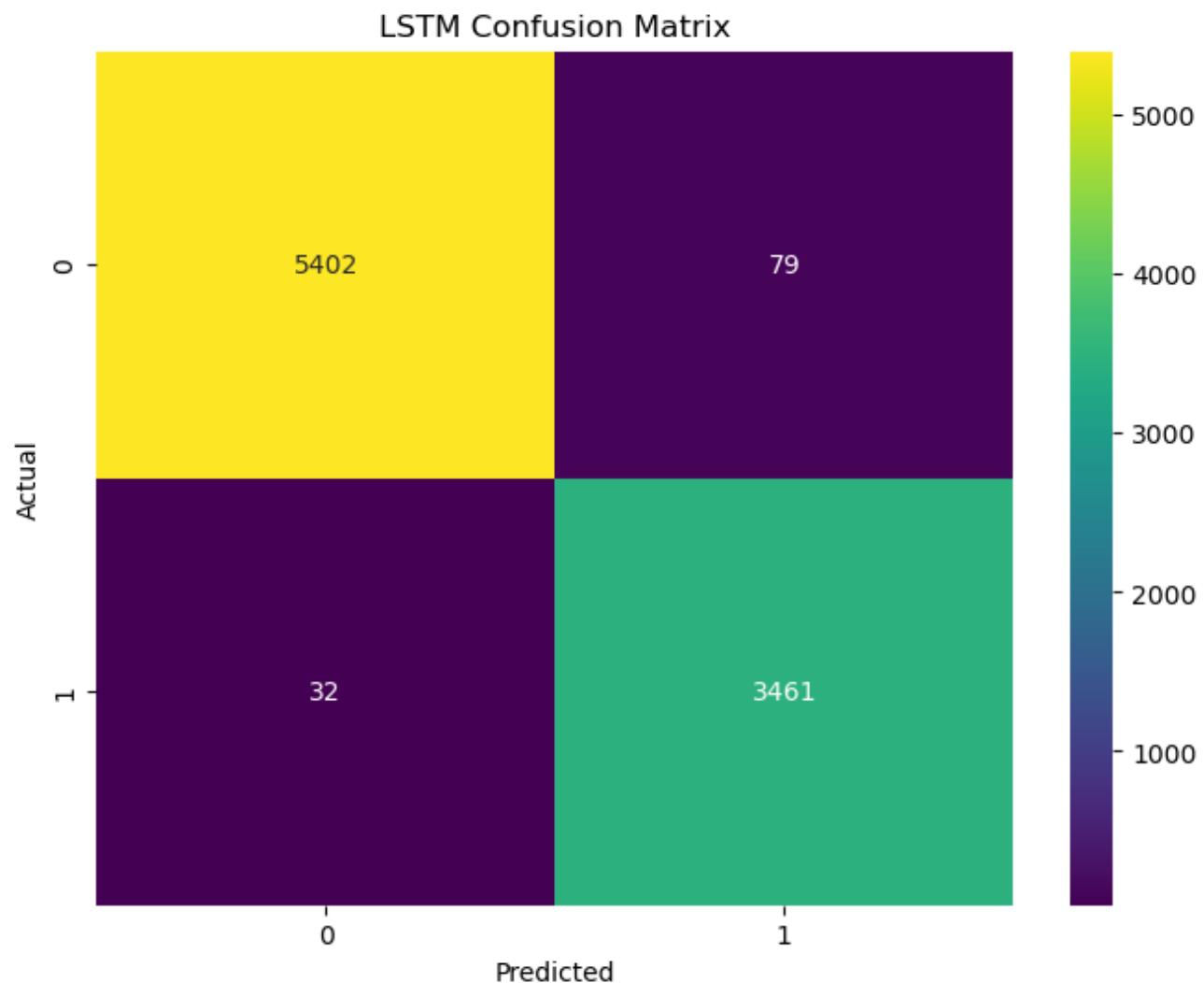


```
In [95]: # Function to visualize confusion matrix
def plot_confusion_matrix(y_true, y_pred, title):
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='g', cmap='viridis')
    plt.title(title)
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.show()
```

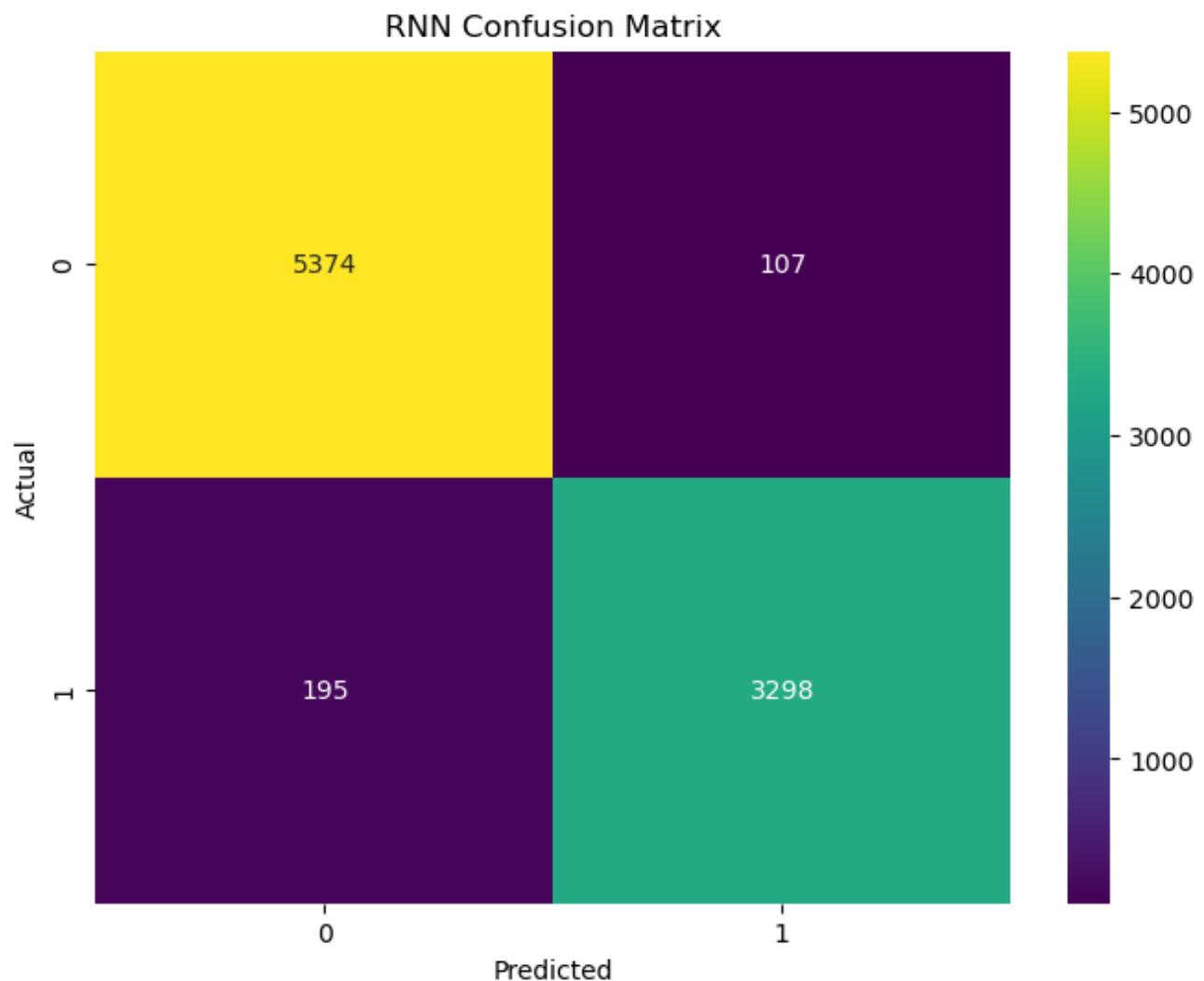
```
In [96]: # Logistic Regression
plot_confusion_matrix(y_val, y_pred_logistic, 'Logistic Regression Confusion
```



```
In [97]: # LSTM  
y_pred_lstm = (y_pred_lstm > 0.5).astype("int32").flatten()  
plot_confusion_matrix(y_val, y_pred_lstm, 'LSTM Confusion Matrix')
```



```
In [98]: # RNN  
y_pred_rnn = (y_pred_rnn > 0.5).astype("int32").flatten()  
plot_confusion_matrix(y_val, y_pred_rnn, 'RNN Confusion Matrix')
```



```
In [153]: # Accuracy Breakdown Analysis by Prompt Name

# Add predictions to validation data
val_data['y_pred_lstm'] = y_pred_lstm
val_data['y_pred_rnn'] = y_pred_rnn

# Group by prompt name and calculate accuracy
prompt_accuracy_lstm = val_data.groupby('prompt_name').apply(lambda x: accuracy_score(x['label'], x['y_pred_lstm']))
prompt_accuracy_rnn = val_data.groupby('prompt_name').apply(lambda x: accuracy_score(x['label'], x['y_pred_rnn']))

print('LSTM Accuracy by Prompt Name:')
print(prompt_accuracy_lstm)

print('\n')

print('RNN Accuracy by Prompt Name:')
print(prompt_accuracy_rnn)
```

LSTM Accuracy by Prompt Name:

prompt_name	
"A Cowboy Who Rode the Waves"	0.986450
Car-free cities	0.997967
Cell phones at school	0.995349
Community service	0.981609
Distance learning	0.980072
Does the electoral college work?	0.994246
Driverless cars	0.986755
Exploring Venus	0.997658
Facial action coding system	0.994718
Grades for extracurricular activities	0.980998
Mandatory extracurricular activities	0.970497
Phones and driving	0.993884
Seeking multiple opinions	0.984344
Summer projects	0.985612
The Face on Mars	0.989041

RNN Accuracy by Prompt Name:

prompt_name	
"A Cowboy Who Rode the Waves"	0.972900
Car-free cities	0.972561
Cell phones at school	0.969767
Community service	0.965517
Distance learning	0.937500
Does the electoral college work?	0.985040
Driverless cars	0.980132
Exploring Venus	0.962529
Facial action coding system	0.980634
Grades for extracurricular activities	0.971496
Mandatory extracurricular activities	0.948758
Phones and driving	0.963303
Seeking multiple opinions	0.968689
Summer projects	0.967626
The Face on Mars	0.967123

```
/var/folders/2x/j2yg_98j10n2xd364kzb8nvm0000gn/T/ipykernel_92515/1772376208.py:8: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explicitly select the grouping columns after groupby to silence this warning.  
    prompt_accuracy_lstm = val_data.groupby('prompt_name').apply(lambda x: accuracy_score(x['label'], x['y_pred_lstm']))  
/var/folders/2x/j2yg_98j10n2xd364kzb8nvm0000gn/T/ipykernel_92515/1772376208.py:9: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explicitly select the grouping columns after groupby to silence this warning.  
    prompt_accuracy_rnn = val_data.groupby('prompt_name').apply(lambda x: accuracy_score(x['label'], x['y_pred_rnn']))
```

```
In [154...]: # Group by source and calculate accuracy  
source_accuracy_lstm = val_data.groupby('source').apply(lambda x: accuracy_score(x['label'], x['y_pred_lstm']))  
source_accuracy_rnn = val_data.groupby('source').apply(lambda x: accuracy_score(x['label'], x['y_pred_rnn']), include_groups=False)  
print('LSTM Accuracy by Source: ')  
print(source_accuracy_lstm)  
  
print('\n')  
  
print('RNN Accuracy by Source: ')  
print(source_accuracy_rnn)
```

LSTM Accuracy by Source:

```
source
NousResearch/Llama-2-7b-chat-hf      1.000000
chat_gpt_moth                         0.995772
cohere-command                         0.915493
darragh_claude_v6                     1.000000
darragh_claude_v7                     1.000000
falcon_180b_v1                         1.000000
kingki19_palm                          1.000000
llama2_chat                            0.998004
llama_70b_v1                           0.982456
mistral7binstruct_v1                  0.993644
mistral7binstruct_v2                  0.968064
mistralai/Mistral-7B-Instruct-v0.1   1.000000
palm-text-bison1                      1.000000
persuade_corpus                        0.985563
radek_500                             1.000000
radekgpt4                            1.000000
train_essays                          0.986014
dtype: float64
```

RNN Accuracy by Source:

```
source
NousResearch/Llama-2-7b-chat-hf      0.970588
chat_gpt_moth                         0.947146
cohere-command                         0.802817
darragh_claude_v6                     0.965517
darragh_claude_v7                     0.989418
falcon_180b_v1                         0.912195
kingki19_palm                          0.988462
llama2_chat                            0.942116
llama_70b_v1                           0.868421
mistral7binstruct_v1                  0.980932
mistral7binstruct_v2                  0.928144
mistralai/Mistral-7B-Instruct-v0.1   0.941176
palm-text-bison1                      0.821429
persuade_corpus                        0.981328
radek_500                             1.000000
radekgpt4                            1.000000
train_essays                          0.965035
dtype: float64
```

```
/var/folders/2x/j2yg_98j10n2xd364kzb8nvm0000gn/T/ipykernel_92515/308316122.p
y:2: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping col
umns. This behavior is deprecated, and in a future version of pandas the gro
uping columns will be excluded from the operation. Either pass `include_grou
ps=False` to exclude the groupings or explicitly select the grouping columns
after groupby to silence this warning.
    source_accuracy_lstm = val_data.groupby('source').apply(lambda x: accuracy_
_score(x['label'], x['y_pred_lstm']))
/var/folders/2x/j2yg_98j10n2xd364kzb8nvm0000gn/T/ipykernel_92515/308316122.p
y:3: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping col
umns. This behavior is deprecated, and in a future version of pandas the gro
uping columns will be excluded from the operation. Either pass `include_grou
ps=False` to exclude the groupings or explicitly select the grouping columns
after groupby to silence this warning.
    source_accuracy_rnn = val_data.groupby('source').apply(lambda x: accuracy_
score(x['label'], x['y_pred_rnn']))
```

```
In [155...]: # Calculate error rate (1 - accuracy)
prompt_error_lstm = 1 - prompt_accuracy_lstm
prompt_error_rnn = 1 - prompt_accuracy_rnn

# Convert the error data to DataFrames
prompt_error_lstm_df = prompt_error_lstm.reset_index()
prompt_error_rnn_df = prompt_error_rnn.reset_index()

# Rename the error column to 'error_rate'
prompt_error_lstm_df.columns = ['prompt_name', 'error_rate']
prompt_error_rnn_df.columns = ['prompt_name', 'error_rate']

# Print column names to verify
print(prompt_error_lstm_df.columns)
print(prompt_error_rnn_df.columns)
```

```
Index(['prompt_name', 'error_rate'], dtype='object')
Index(['prompt_name', 'error_rate'], dtype='object')
```

```
In [163...]: plt.figure(figsize=(14, 8))

# Plot LSTM error rate
plt.subplot(1, 2, 1)
sns.barplot(x='error_rate', y='prompt_name', data=prompt_error_lstm_df, palette='p
plt.title('LSTM Error Rate by Prompt Name')
plt.xlabel('Error Rate')
plt.ylabel('Prompt Name')
plt.xlim(0, 0.1) # Set x-axis limit to 1 for clarity

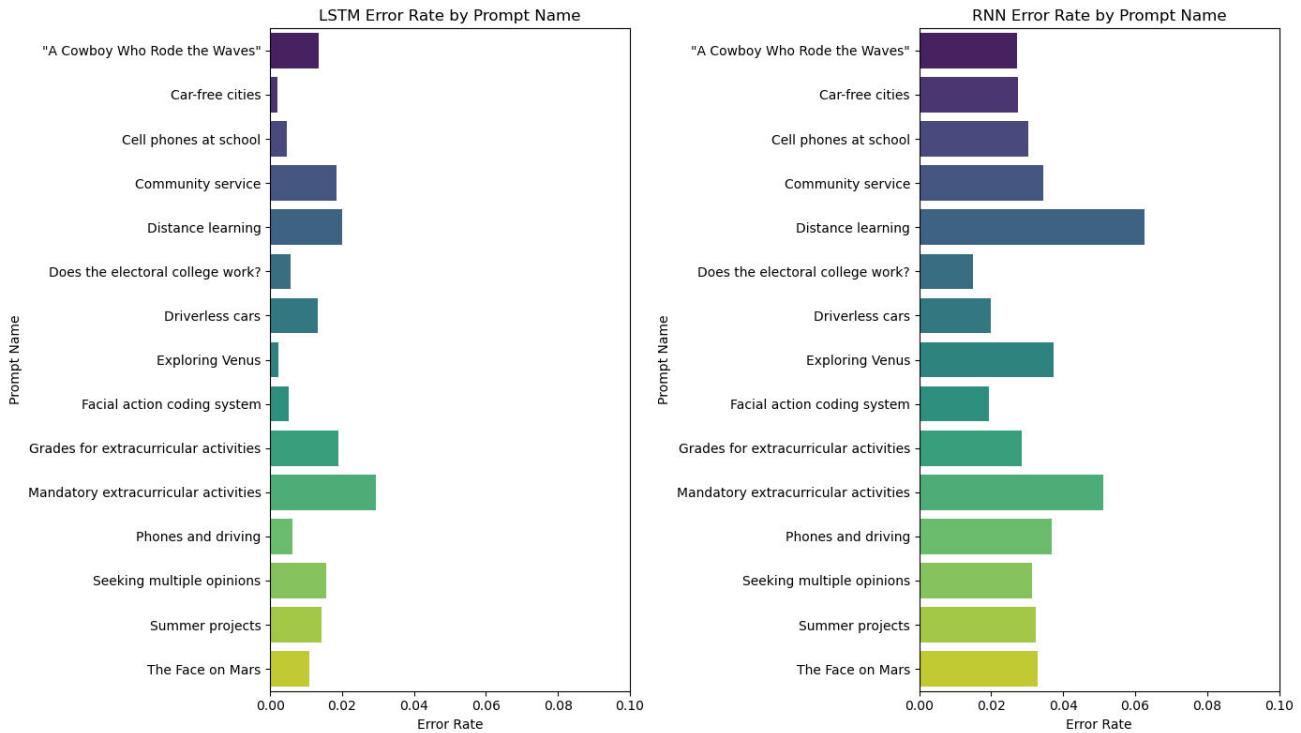
# Plot RNN error rate
plt.subplot(1, 2, 2)
sns.barplot(x='error_rate', y='prompt_name', data=prompt_error_rnn_df, palette='p
```

```

plt.xlabel('Error Rate')
plt.ylabel('Prompt Name')
plt.xlim(0, 0.1) # Set x-axis limit to 1 for clarity

# Adjust layout
plt.tight_layout()
plt.show()

```



In [142]: # Build the Bidirectional LSTM model

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, Bidirectional, LSTM, Dense, Dropout

model_bi_lstm = Sequential()
model_bi_lstm.add(Embedding(max_features, 128, input_length=maxlen))
model_bi_lstm.add(Bidirectional(LSTM(64, return_sequences=True)))
model_bi_lstm.add(Dropout(0.5))
model_bi_lstm.add(Bidirectional(LSTM(64)))
model_bi_lstm.add(Dropout(0.5))
model_bi_lstm.add(Dense(1, activation='sigmoid'))

model_bi_lstm.compile(optimizer='adam', loss='binary_crossentropy', metrics=[accuracy])

# Train the model
bi_lstm_history = model_bi_lstm.fit(X_train, y_train, epochs=5, batch_size=32)

```

2024-08-07 12:01:20.888693: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must fe

```
ed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:01:20.889163: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:01:20.889705: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 12:01:20.931738: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis' with dtype int32 and shape [1]
[[{{node gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis)}}]]
2024-08-07 12:01:20.944608: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:01:20.945027: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:01:20.945373: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 12:01:21.006753: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:01:21.007102: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
```

```
[[{{node gradients/split_grad(concat/split/split_dim)}}]]  
2024-08-07 12:01:21.007472: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32  
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]  
Epoch 1/5  
2024-08-07 12:01:21.325714: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis' with dtype int32 and shape [1]  
[[{{node gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis)}}]]  
2024-08-07 12:01:21.339189: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32  
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]  
2024-08-07 12:01:21.339619: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32  
[[{{node gradients/split_grad(concat/split/split_dim)}}]]  
2024-08-07 12:01:21.340007: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32  
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]  
2024-08-07 12:01:21.424012: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32  
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]  
2024-08-07 12:01:21.424509: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32  
[[{{node gradients/split_grad(concat/split/split_dim)}}]]  
2024-08-07 12:01:21.424916: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32  
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
```

```
2024-08-07 12:01:21.467411: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/ReverseV2_grad/ReverseV2/Revers  
eV2/axis' with dtype int32 and shape [1]  
    [[{{node gradients/ReverseV2_grad/ReverseV2/ReverseV2/axis}}]]  
2024-08-07 12:01:21.481048: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]  
2024-08-07 12:01:21.481458: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_grad(concat/split/split_d  
im' with dtype int32  
    [[{{node gradients/split_grad(concat/split/split_dim)}}]]  
2024-08-07 12:01:21.481850: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]  
2024-08-07 12:01:21.541428: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]  
2024-08-07 12:01:21.541933: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_grad(concat/split/split_d  
im' with dtype int32  
    [[{{node gradients/split_grad(concat/split/split_dim)}}]]  
2024-08-07 12:01:21.542451: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]  
2024-08-07 12:01:21.586362: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/ReverseV2_grad/ReverseV2/Revers  
eV2/axis' with dtype int32 and shape [1]  
    [[{{node gradients/ReverseV2_grad/ReverseV2/ReverseV2/axis}}]]  
2024-08-07 12:01:21.600166: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic
```

```
ate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:01:21.600643: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:01:21.601050: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 12:01:21.746874: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis' with dtype int32 and shape [1]
[[{{node gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis)}}]]
2024-08-07 12:01:21.872998: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis' with dtype int32 and shape [1]
[[{{node gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis)}}]]
2024-08-07 12:01:21.973950: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:01:21.974602: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:01:21.974971: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 12:01:22.015810: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis'
```

```
eV2/axis' with dtype int32 and shape [1]
[[{{node gradients/ReverseV2_grad/ReverseV2/ReverseV2/axis}}]]
2024-08-07 12:01:22.028263: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:01:22.028704: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:01:22.029023: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 12:01:22.082035: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:01:22.082435: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:01:22.082786: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 12:01:22.121717: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/ReverseV2_grad/ReverseV2/ReverseV2/axis' with dtype int32 and shape [1]
[[{{node gradients/ReverseV2_grad/ReverseV2/ReverseV2/axis}}]]
2024-08-07 12:01:22.134543: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
```

```
2024-08-07 12:01:22.134976: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_grad/concat/split/split_d  
im' with dtype int32  
    [[{{node gradients/split_grad/concat/split/split_dim}}]]  
2024-08-07 12:01:22.135388: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_1_grad/concat/split_1/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_1_grad/concat/split_1/split_dim}}]]  
2024-08-07 12:01:22.272205: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/ReverseV2_grad/ReverseV2/Revers  
eV2/axis' with dtype int32 and shape [1]  
    [[{{node gradients/ReverseV2_grad/ReverseV2/ReverseV2/axis}}]]  
2024-08-07 12:01:22.386881: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/ReverseV2_grad/ReverseV2/Revers  
eV2/axis' with dtype int32 and shape [1]  
    [[{{node gradients/ReverseV2_grad/ReverseV2/ReverseV2/axis}}]]  
1122/1122 [=====] - ETA: 0s - loss: 0.0916 - accura  
cy: 0.9689  
2024-08-07 12:02:58.481159: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_2_grad/concat/split_2/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_2_grad/concat/split_2/split_dim}}]]  
2024-08-07 12:02:58.481677: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_grad/concat/split/split_d  
im' with dtype int32  
    [[{{node gradients/split_grad/concat/split/split_dim}}]]  
2024-08-07 12:02:58.482053: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_1_grad/concat/split_1/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_1_grad/concat/split_1/split_dim}}]]  
2024-08-07 12:02:58.523656: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/ReverseV2_grad/ReverseV2/Revers  
eV2/axis' with dtype int32 and shape [1]  
    [[{{node gradients/ReverseV2_grad/ReverseV2/ReverseV2/axis}}]]
```

```
2024-08-07 12:02:58.536038: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]  
2024-08-07 12:02:58.536367: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_grad(concat/split/split_d  
im' with dtype int32  
    [[{{node gradients/split_grad(concat/split/split_dim)}}]]  
2024-08-07 12:02:58.536726: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]  
2024-08-07 12:02:58.589779: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]  
2024-08-07 12:02:58.590201: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_grad(concat/split/split_d  
im' with dtype int32  
    [[{{node gradients/split_grad(concat/split/split_dim)}}]]  
2024-08-07 12:02:58.590566: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]  
2024-08-07 12:02:58.630067: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/ReverseV2_grad/ReverseV2/Revers  
eV2/axis' with dtype int32 and shape [1]  
    [[{{node gradients/ReverseV2_grad/ReverseV2/ReverseV2/axis}}]]  
2024-08-07 12:02:58.642355: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic  
ate an error and you can ignore this message): INVALID_ARGUMENT: You must fe  
ed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/spl  
it_dim' with dtype int32  
    [[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]  
2024-08-07 12:02:58.642748: I tensorflow/core/common_runtime/executor.cc:119  
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indic
```

```
ate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim)' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:02:58.643110: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim)' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
1122/1122 [=====] - 104s 91ms/step - loss: 0.0916 - accuracy: 0.9689 - val_loss: 0.0560 - val_accuracy: 0.9837
Epoch 2/5
1122/1122 [=====] - 102s 91ms/step - loss: 0.0357 - accuracy: 0.9904 - val_loss: 0.0532 - val_accuracy: 0.9824
Epoch 3/5
1122/1122 [=====] - 102s 91ms/step - loss: 0.0224 - accuracy: 0.9941 - val_loss: 0.0336 - val_accuracy: 0.9899
Epoch 4/5
1122/1122 [=====] - 102s 91ms/step - loss: 0.0150 - accuracy: 0.9960 - val_loss: 0.0485 - val_accuracy: 0.9873
Epoch 5/5
1122/1122 [=====] - 101s 90ms/step - loss: 0.0183 - accuracy: 0.9948 - val_loss: 0.0436 - val_accuracy: 0.9899
```

In [143...]

```
# Make predictions on the validation set
y_pred_bi_lstm = (model_bi_lstm.predict(X_val) > 0.5).astype("int32")

# Calculate accuracy
bi_lstm_accuracy = accuracy_score(y_val, y_pred_bi_lstm)
print(f'Bidirectional LSTM Validation Accuracy: {bi_lstm_accuracy:.4f}')

# Classification report
print('Classification Report:')
print(classification_report(y_val, y_pred_bi_lstm))
```

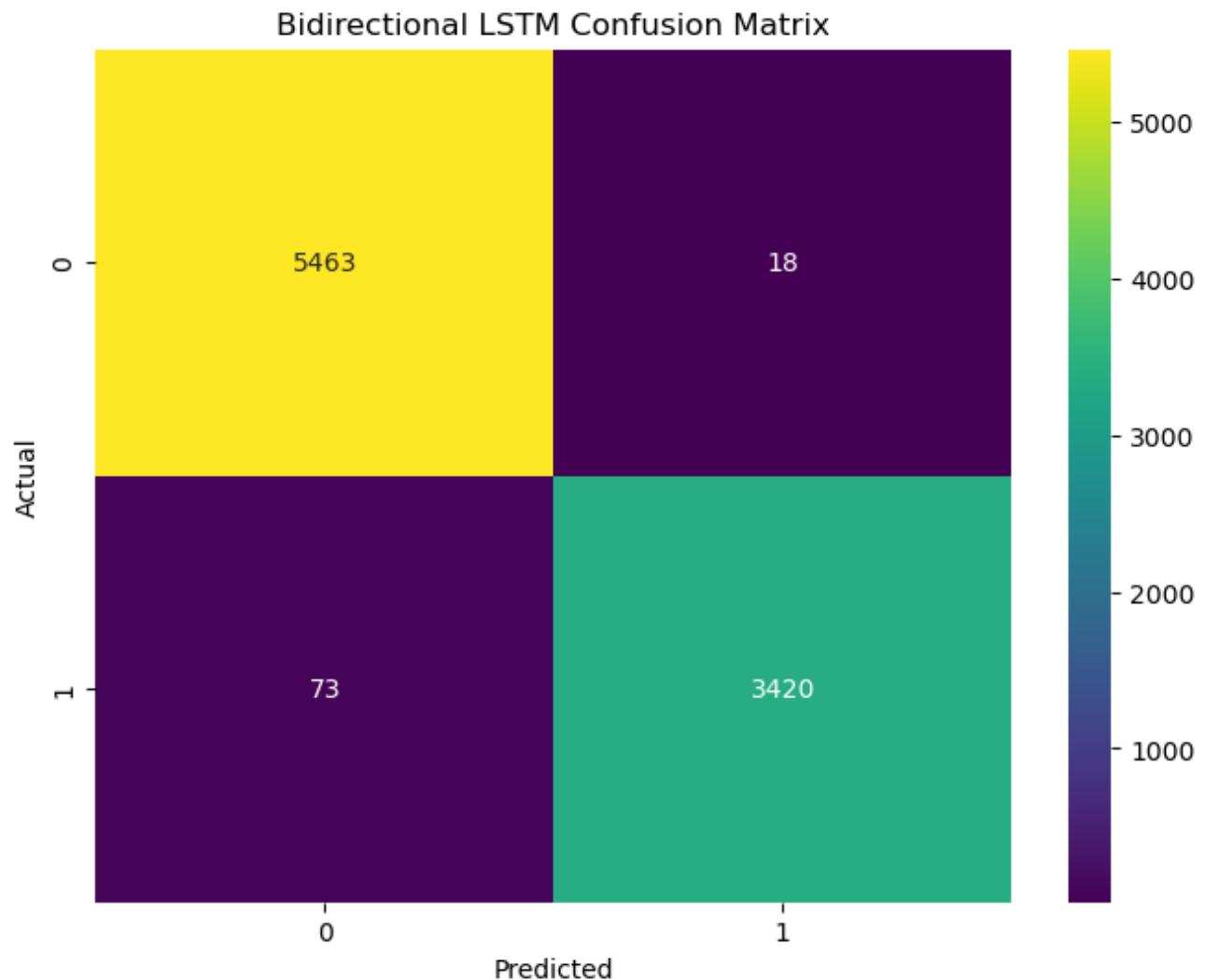
```
2024-08-07 12:09:51.868288: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim)' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:09:51.868755: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim)' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:09:51.869353: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate
```

```
ate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 12:09:51.909454: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis' with dtype int32 and shape [1]
[[{{node gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis)}}]]
2024-08-07 12:09:51.921978: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:09:51.922317: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:09:51.922675: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 12:09:51.974433: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:09:51.974823: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:09:51.975232: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 12:09:52.015544: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis'
```

```
eV2/axis' with dtype int32 and shape [1]
[[{{node gradients/ReverseV2_grad/ReverseV2/ReverseV2/axis}}]]
2024-08-07 12:09:52.027972: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:09:52.028398: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:09:52.028794: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
281/281 [=====] - 7s 22ms/step
Bidirectional LSTM Validation Accuracy: 0.9899
Classification Report:
precision    recall    f1-score    support
0           0.99      1.00      0.99      5481
1           0.99      0.98      0.99      3493

accuracy                           0.99      8974
macro avg       0.99      0.99      0.99      8974
weighted avg    0.99      0.99      0.99      8974
```

```
In [144]: # Confusion matrix
y_pred_bi_lstm = (y_pred_bi_lstm > 0.5).astype("int32").flatten()
plot_confusion_matrix(y_val, y_pred_bi_lstm, 'Bidirectional LSTM Confusion M
```

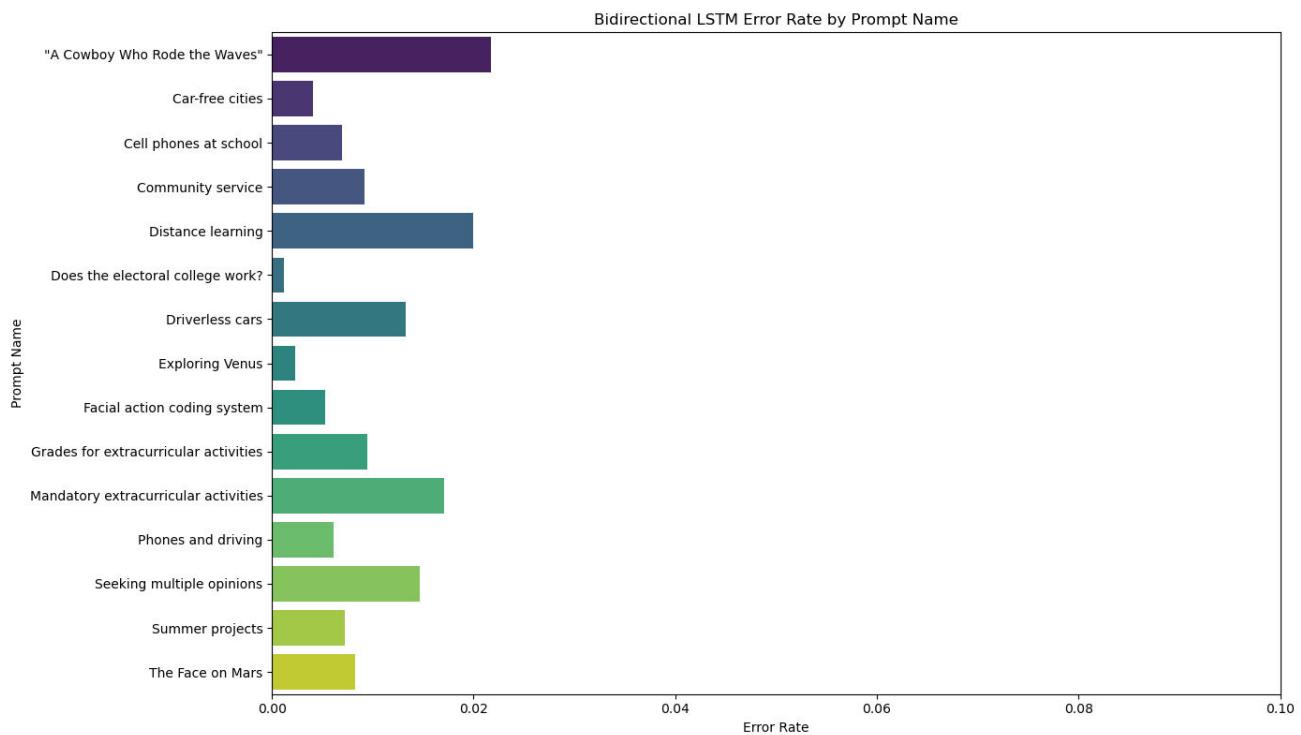


```
In [115]: # Save the Bidirectional LSTM  
model_bi_lstm.save('bi_lstm_model.h5')
```

```
In [164]: # Group by prompt name and calculate accuracy  
val_data['y_pred_bi_lstm'] = y_pred_bi_lstm  
prompt_accuracy_bi_lstm = val_data.groupby('prompt_name').apply(lambda x: acc(x))  
  
# Calculate error rate (1 - accuracy)  
prompt_error_bi_lstm = 1 - prompt_accuracy_bi_lstm  
  
# Convert the error data to DataFrame  
prompt_error_bi_lstm_df = prompt_error_bi_lstm.reset_index()  
prompt_error_bi_lstm_df.columns = ['prompt_name', 'error_rate']  
  
# Plot the error rates  
plt.figure(figsize=(14, 8))  
  
# Plot Bidirectional RNN error rate  
sns.barplot(x='error_rate', y='prompt_name', data=prompt_error_bi_lstm_df, p
```

```
plt.title('Bidirectional LSTM Error Rate by Prompt Name')
plt.xlabel('Error Rate')
plt.ylabel('Prompt Name')
plt.xlim(0, 0.1) # Set x-axis limit to 1 for clarity

# Adjust layout
plt.tight_layout()
plt.show()
```



In [17]: # Appendix II – Text Testing Widget Application

```
import ipywidgets as widgets
from IPython.display import display, HTML
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.sequence import pad_sequences
import pickle

# Load the LSTM model and tokenizer
model_lstm = tf.keras.models.load_model('bi_lstm_model.h5')
with open('tokenizer.pkl', 'rb') as handle:
    tokenizer = pickle.load(handle)

# Create a text area widget
text_area = widgets.Textarea(
    value='',
    placeholder='Insert Essay Here',
    description='Text Here:',
    disabled=False,
    layout=widgets.Layout(width='100%', height='300px')
)

# Create an output area
output = widgets.Output()

# Define the function to make predictions
def make_prediction(change):
    text = change['new']

    # Tokenize and pad the text
    text_seq = tokenizer.texts_to_sequences([text])
    text_pad = pad_sequences(text_seq, maxlen=150) # Same max length as in

    # Make prediction
    prediction_prob = model_lstm.predict(text_pad)[0][0]
    prediction = 'AI-written' if prediction_prob > 0.5 else 'Human-written'
    confidence = prediction_prob if prediction == 'AI-written' else 1 - pre

    # Result display
    result = {
        'prediction': prediction,
        'confidence': f'{confidence * 100:.2f}%'
    }
    with output:
        output.clear_output()
        display(HTML(f"<p><strong>Prediction:</strong> {result['prediction']}
```

```
# Link the text area widget to the function
text_area.observe(make_prediction, names='value')

# Display the widgets
display(text_area, output)
```

```
2024-08-07 12:24:05.535038: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:24:05.535490: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:24:05.536114: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 12:24:05.577388: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis)' with dtype int32 and shape [1]
[[{{node gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis)}}]]
2024-08-07 12:24:05.589714: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
[[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:24:05.590142: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
[[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:24:05.590505: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
[[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 12:24:05.651823: I tensorflow/core/common_runtime/executor.cc:119
```

```
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
    [[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:24:05.652293: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
    [[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:24:05.652723: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
    [[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
2024-08-07 12:24:05.693186: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis' with dtype int32 and shape [1]
    [[{{node gradients/ReverseV2_grad(ReverseV2/ReverseV2/axis)}}]]
2024-08-07 12:24:05.706085: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_2_grad(concat/split_2/split_dim' with dtype int32
    [[{{node gradients/split_2_grad(concat/split_2/split_dim)}}]]
2024-08-07 12:24:05.706486: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_grad(concat/split/split_dim' with dtype int32
    [[{{node gradients/split_grad(concat/split/split_dim)}}]]
2024-08-07 12:24:05.706847: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'gradients/split_1_grad(concat/split_1/split_dim' with dtype int32
    [[{{node gradients/split_1_grad(concat/split_1/split_dim)}}]]
Textarea(value='', description='Text Here:', layout=Layout(height='300px', width='100%'), placeholder='Insert ...
Output()
```