

FakeNewsNetCNN

September 8, 2024

1 FakeNewsNetCNN

This notebook builds and evaluates Convolutional Neural Network (CNN) model for detecting fake news using the FakeNewsNet dataset, that contains two subsets, GossipCop and PolitiFact. This work is based on the work of Shu et al. [7] and Denny Britz repository. [11]

1.1 Import necessary libraries

```
[4]: import numpy as np
import random
import tensorflow as tf
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import joblib
from sklearn.metrics import accuracy_score, precision_score, recall_score, \
    f1_score, roc_auc_score, confusion_matrix, classification_report, roc_curve
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split, KFold
from sklearn.utils import class_weight
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv1D, MaxPooling1D, \
    GlobalMaxPooling1D, Flatten, Dropout, Embedding
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.regularizers import l2
from tensorflow.keras.backend import clear_session

# Check and install keras-tuner if not available
try:
    import keras_tuner as kt
except ImportError:
    !pip install keras-tuner
    import keras_tuner as kt
```

1.2 Add Reproducibility logic

Functions to set seeds and preserve deterministic operations are defined for reproducibility. Different runs have the same results.

```
[5]: # Set seeds for reproducibility
def set_seeds(seed=42):
    np.random.seed(seed)
    random.seed(seed)
    tf.random.set_seed(seed)

set_seeds()

# Ensure deterministic operations
def set_deterministic():
    tf.config.threading.set_intra_op_parallelism_threads(1)
    tf.config.threading.set_inter_op_parallelism_threads(1)

# Call set_deterministic before any TensorFlow operations
set_deterministic()
```

1.3 Load Preprocessed Data from Google Drive

```
[6]: # Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Load the preprocessed datasets from Google Drive
gossipcop_combined = pd.read_csv('/content/drive/My Drive/
↳gossipcop_preprocessed.csv')
politifact_combined = pd.read_csv('/content/drive/My Drive/
↳politifact_preprocessed.csv')

# Quick check of the data loaded
print(gossipcop_combined.head())
print(politifact_combined.head())
```

Mounted at /content/drive

	title	label
0	lea michel hairstylist mix textur spray coconu...	1
1	thoma markl princ harri polit miss daughter me...	0
2	2019 sag award nomin see full list nomine varieti	1
3	see megan markl royal coat arm symbol hide wi...	1
4	kyli jenner visit shaman life kyli season final	1

	title	label
0	world popular candi remov shelv octob 2017	0
1	brows congression bill	1
2	suprem court vacanc video	1

```

3                                     u import export          1
4  die 78 year old cia agent admit kill marilyn m...          0

```

1.4 Check loaded data

```

[7]: # Check for NaN values in datasets
print("NaN values in GossipCop dataset:\n", gossipcop_combined.isnull().sum())
print("NaN values in PolitiFact dataset:\n", politifact_combined.isnull().sum())

# Ensure there are no NaN values
gossipcop_combined.dropna(inplace=True)
politifact_combined.dropna(inplace=True)

# Re-check for NaN values in datasets
print("NaN values in GossipCop dataset after dropna:\n", gossipcop_combined.
      ↪isnull().sum())
print("NaN values in PolitiFact dataset after dropna:\n", politifact_combined.
      ↪isnull().sum())

```

```

NaN values in GossipCop dataset:
  title    1
label    0
dtype: int64
NaN values in PolitiFact dataset:
  title    2
label    0
dtype: int64
NaN values in GossipCop dataset after dropna:
  title    0
label    0
dtype: int64
NaN values in PolitiFact dataset after dropna:
  title    0
label    0
dtype: int64

```

1.5 Prepare Data for CNN

Features (X) and labels (y) are defined for both datasets. Text data is tokenized, padded, and split into training and testing sets. Tokenization converts text into numerical values. This process makes it suitable for neural network operations. Padding keeps uniform input length and enables efficient batch processing as explained by Denny Britz blog post linked to his repository[11]. Encoding labels into numerical form standardizes the output for classification tasks. I am splitting the dataset the same way as Shu et al. [7]: “We use 80% of data for training and 20% for testing.”

```

[8]: # Define features (X) and labels (y)
X_gossipcop = gossipcop_combined['title'].values
y_gossipcop = gossipcop_combined['label'].values

```

```

X_politifact = politifact_combined['title'].values
y_politifact = politifact_combined['label'].values

# Function to tokenize and pad sequences
def tokenize_and_pad(texts, max_num_words=5000, maxlen=100):
    tokenizer = Tokenizer(num_words=max_num_words)
    tokenizer.fit_on_texts(texts)
    sequences = tokenizer.texts_to_sequences(texts)
    padded_sequences = pad_sequences(sequences, padding='post', maxlen=maxlen)
    return padded_sequences, tokenizer

# Tokenize and pad the sequences for both datasets
X_gossipcop, tokenizer_gossipcop = tokenize_and_pad(X_gossipcop)
X_politifact, tokenizer_politifact = tokenize_and_pad(X_politifact)

# Encode the labels as integers
def encode_labels(labels):
    le = LabelEncoder()
    encoded_labels = le.fit_transform(labels)
    return encoded_labels, le

y_gossipcop, le_gossipcop = encode_labels(y_gossipcop)
y_politifact, le_politifact = encode_labels(y_politifact)

# Split the data into training and testing sets for both datasets with a test
↳size of 20%
def split_data(X, y, test_size=0.2, random_state=42):
    return train_test_split(X, y, test_size=test_size,
↳random_state=random_state)

X_train_gossipcop, X_test_gossipcop, y_train_gossipcop, y_test_gossipcop =
↳split_data(X_gossipcop, y_gossipcop)
X_train_politifact, X_test_politifact, y_train_politifact, y_test_politifact =
↳split_data(X_politifact, y_politifact)

```

1.6 Build the CNN Model

Separate CNN models for GossipCop and PolitiFact are built using embedding, convolutional, pooling, dropout, and dense layers. Models are compiled with Adam optimizer and binary crossentropy loss. I have experimented with different layers, parameters and settings explained by Denny Britz [11].

```

[9]: # Build the CNN Model for GossipCop
def create_cnn_model_gossipcop(input_length):
    model = Sequential([
        Embedding(input_dim=5000, output_dim=128, input_length=input_length),
↳# Embedding layer

```

```

        Conv1D(filters=64, kernel_size=5, activation='relu',
↳kernel_regularizer=l2(0.01)), # Conv1D layer
        MaxPooling1D(pool_size=2), # MaxPooling layer
        Conv1D(filters=32, kernel_size=5, activation='relu',
↳kernel_regularizer=l2(0.01)), # Conv1D layer
        GlobalMaxPooling1D(), # GlobalMaxPooling layer
        Dense(units=32, activation='relu'), # Dense layer
        Dropout(0.5), # Dropout layer
        Dense(units=1, activation='sigmoid') # Output layer
    ])
    model.compile(optimizer=Adam(learning_rate=0.0001),
↳loss='binary_crossentropy', metrics=['accuracy'])
    return model

# Build the CNN Model for PolitiFact
def create_cnn_model_politifact(input_length):
    model = Sequential([
        Embedding(input_dim=5000, output_dim=128, input_length=input_length),
↳# Embedding layer
        Conv1D(filters=128, kernel_size=5, activation='relu',
↳kernel_regularizer=l2(0.01)), # Conv1D layer
        MaxPooling1D(pool_size=2), # MaxPooling layer
        Conv1D(filters=64, kernel_size=5, activation='relu',
↳kernel_regularizer=l2(0.01)), # Conv1D layer
        GlobalMaxPooling1D(), # GlobalMaxPooling layer
        Dense(units=64, activation='relu', kernel_regularizer=l2(0.01)), #
↳Dense layer
        Dropout(0.6), # Dropout layer
        Dense(units=1, activation='sigmoid', kernel_regularizer=l2(0.01)) #
↳Output layer
    ])
    model.compile(optimizer=Adam(learning_rate=0.0001),
↳loss='binary_crossentropy', metrics=['accuracy'])
    return model

input_length = X_train_gossipcop.shape[1]

cnn_model_gossipcop = create_cnn_model_gossipcop(input_length)
cnn_model_politifact = create_cnn_model_politifact(input_length)

# Callbacks to prevent overfitting and make training more efficient
early_stopping_gossipcop = EarlyStopping(monitor='val_loss', patience=3,
↳restore_best_weights=True) # Stop training when a monitored metric has
↳stopped improving
early_stopping_politifact = EarlyStopping(monitor='val_loss', patience=7,
↳restore_best_weights=True)

```

```
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=3,
    ↪min_lr=0.00001) # Reduce the learning rate when a metric has stopped
    ↪improving
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
    warnings.warn(
```

1.7 Handle Class Imbalance

Class weights are calculated to handle class imbalance in the datasets. These weights are used during model training to balance the impact of each class.

```
[10]: # Function to compute class weights and return as a dictionary
def compute_class_weights(y_train):
    class_weights = class_weight.compute_class_weight(class_weight='balanced',
    ↪classes=np.unique(y_train), y=y_train)
    class_weights_dict = {i: class_weights[i] for i in
    ↪range(len(class_weights))}
    return class_weights_dict

# Calculate class weights for GossipCop
class_weights_dict_gossipcop = compute_class_weights(y_train_gossipcop)

# Calculate class weights for PolitiFact
class_weights_dict_politifact = compute_class_weights(y_train_politifact)
```

1.8 Train and Evaluate CNN Model

The CNN models are trained with early stopping and learning rate reduction callbacks. Performance metrics accuracy, precision, recall, F1-score, and ROC AUC are calculated and displayed, also with confusion matrices and ROC curves.

According to Powers et al.[16], “The F1-score, which is the harmonic mean of precision and recall, is particularly recommended for imbalanced datasets because it provides a balance between false positives and false negatives, thus giving a more comprehensive measure of a model’s performance.” so I will use F1 score during training and evaluation.

Keras doesn’t natively support the F1 score as a metric during training. I will keep accuracy as a metric for monitoring during training because it’s fast and gives a general sense of model performance. I will use a custom callback to log the F1 score at the end of each epoch.

```
[11]: # Custom callback to log the F1 score
class F1ScoreCallback(tf.keras.callbacks.Callback):
    def __init__(self, validation_data):
        self.validation_data = validation_data
        self.f1_scores = []
        self.precision_scores = []
        self.recall_scores = []
```

```

def on_epoch_end(self, epoch, logs=None):
    val_pred = (self.model.predict(self.validation_data[0]) > 0.5).
    ↪astype("int32")
    val_true = self.validation_data[1]
    val_f1 = f1_score(val_true, val_pred)
    val_precision = precision_score(val_true, val_pred)
    val_recall = recall_score(val_true, val_pred)
    self.f1_scores.append(val_f1)
    self.precision_scores.append(val_precision)
    self.recall_scores.append(val_recall)
    print(f' - val_f1: {val_f1:.4f} - val_precision: {val_precision:.4f} -
    ↪val_recall: {val_recall:.4f}')

```

```

[12]: # Train and evaluate the model, returning the model, predictions and history
def train_and_evaluate(model, X_train, X_test, y_train, y_test, class_weights,
    ↪early_stopping, reduce_lr, train_model=True):
    f1_callback = F1ScoreCallback(validation_data=(X_test, y_test))
    history = None

    if train_model:
        # Train the model
        history = model.fit(X_train, y_train, epochs=50, batch_size=64,
            validation_split=0.2, class_weight=class_weights,
    ↪verbose=1,
            callbacks=[early_stopping, reduce_lr, f1_callback])

        # Update history to include F1, precision, and recall
        history.history['val_f1_score'] = f1_callback.f1_scores
        history.history['val_precision'] = f1_callback.precision_scores
        history.history['val_recall'] = f1_callback.recall_scores

    # Make predictions
    y_pred_prob = model.predict(X_test)
    y_pred = (y_pred_prob > 0.5).astype("int32")

    # Calculate metrics
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    roc_auc = roc_auc_score(y_test, y_pred_prob)

    # Print and plot evaluation metrics and figures
    print(f'Accuracy: {accuracy:.4f}')
    print(f'Precision: {precision:.4f}')
    print(f'Recall: {recall:.4f}')
    print(f'F1-score: {f1:.4f}')

```

```

print(f'ROC AUC: {roc_auc:.4f}')

cm = confusion_matrix(y_test, y_pred)
print('Confusion Matrix:')
print(cm)

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['False', 'True'], yticklabels=['False', 'True'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, label=f'ROC Curve (area = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend(loc="lower right")
plt.show()

return model, y_pred, history

```

1.8.1 Model Training and Evaluation

```

[13]: # Train and evaluate the CNN model for GossipCop dataset
cnn_model_gossipcop, y_pred_gossipcop, history_gossipcop =
    ↪ train_and_evaluate(cnn_model_gossipcop, X_train_gossipcop, X_test_gossipcop,
    ↪ y_train_gossipcop, y_test_gossipcop, class_weights_dict_gossipcop,
    ↪ early_stopping_gossipcop, reduce_lr, train_model=True)

# Train and evaluate the CNN model for PolitiFact dataset
cnn_model_politifact, y_pred_politifact, history_politifact =
    ↪ train_and_evaluate(cnn_model_politifact, X_train_politifact,
    ↪ X_test_politifact, y_train_politifact, y_test_politifact,
    ↪ class_weights_dict_politifact, early_stopping_politifact, reduce_lr,
    ↪ train_model=True)

```

Epoch 1/50

139/139 1s 9ms/step

- val_f1: 0.8683 - val_precision: 0.7683 - val_recall: 0.9982

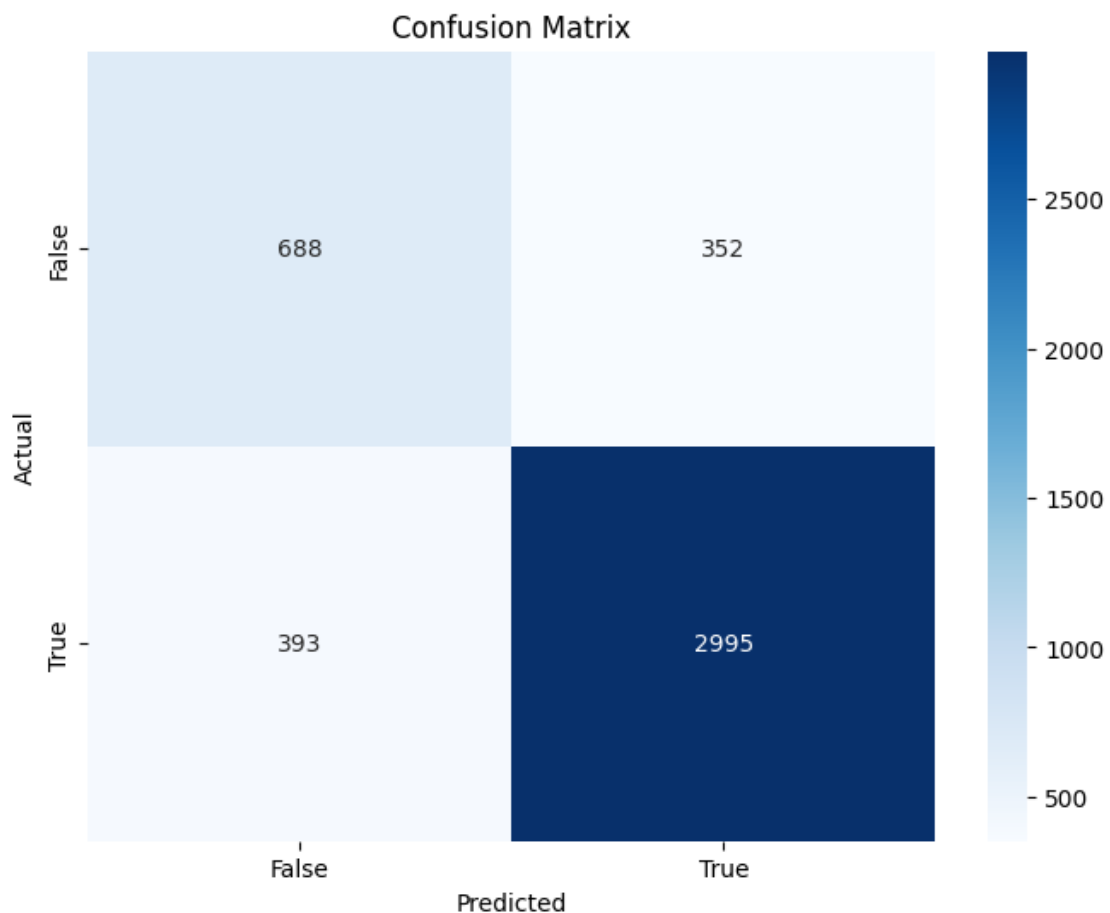
222/222 25s 92ms/step -

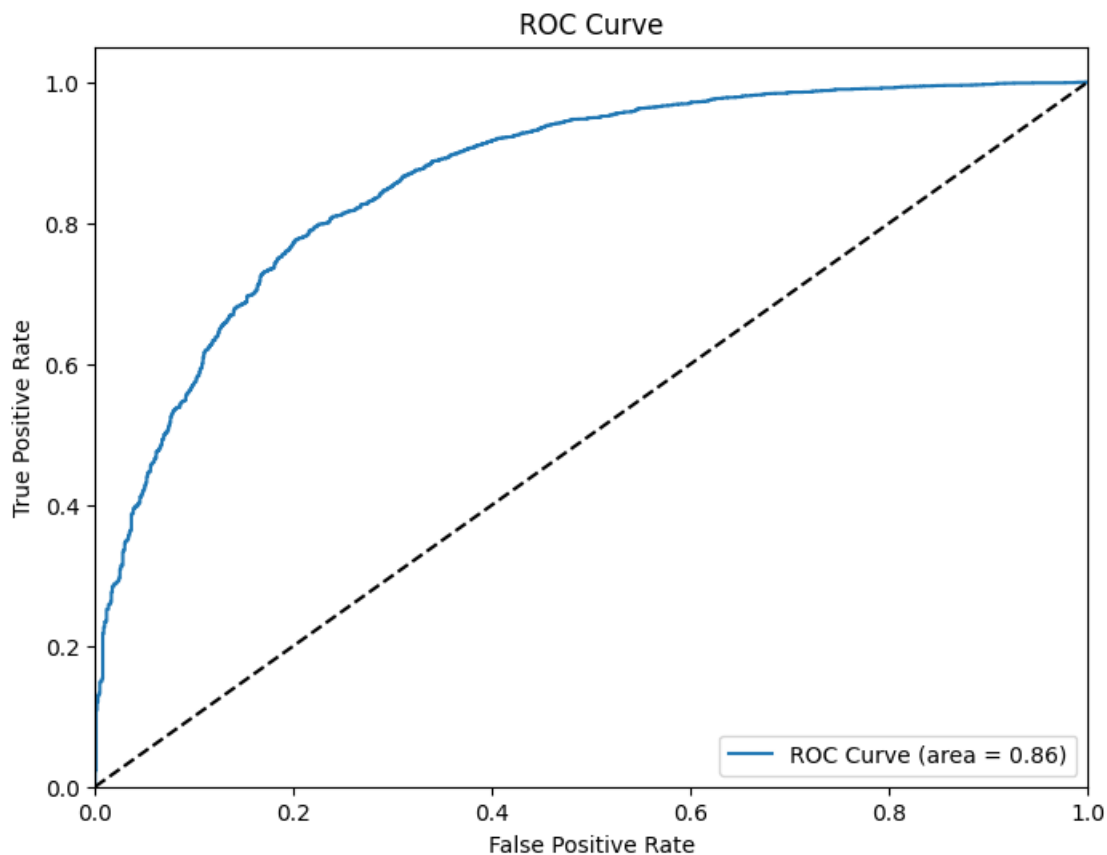

```

accuracy: 0.4799 - loss: 1.7763 - val_accuracy: 0.7671 - val_loss: 1.2929 -
learning_rate: 1.0000e-04
Epoch 2/50
139/139          1s 9ms/step
- val_f1: 0.8825 - val_precision: 0.7996 - val_recall: 0.9847
222/222          18s 81ms/step -
accuracy: 0.6306 - loss: 1.2043 - val_accuracy: 0.8013 - val_loss: 0.9562 -
learning_rate: 1.0000e-04
Epoch 3/50
139/139          1s 9ms/step
- val_f1: 0.8891 - val_precision: 0.8578 - val_recall: 0.9227
222/222          22s 89ms/step -
accuracy: 0.7706 - loss: 0.9178 - val_accuracy: 0.8236 - val_loss: 0.7375 -
learning_rate: 1.0000e-04
Epoch 4/50
139/139          1s 9ms/step
- val_f1: 0.8701 - val_precision: 0.9047 - val_recall: 0.8380
222/222          20s 90ms/step -
accuracy: 0.8028 - loss: 0.7131 - val_accuracy: 0.7959 - val_loss: 0.5666 -
learning_rate: 1.0000e-04
Epoch 5/50
139/139          1s 9ms/step
- val_f1: 0.8815 - val_precision: 0.9008 - val_recall: 0.8630
222/222          18s 82ms/step -
accuracy: 0.8223 - loss: 0.5492 - val_accuracy: 0.8148 - val_loss: 0.5063 -
learning_rate: 1.0000e-04
Epoch 6/50
139/139          2s 13ms/step
- val_f1: 0.8900 - val_precision: 0.8979 - val_recall: 0.8822
222/222          23s 93ms/step -
accuracy: 0.8435 - loss: 0.4742 - val_accuracy: 0.8177 - val_loss: 0.4856 -
learning_rate: 1.0000e-04
Epoch 7/50
139/139          1s 9ms/step
- val_f1: 0.8894 - val_precision: 0.8948 - val_recall: 0.8840
222/222          18s 83ms/step -
accuracy: 0.8602 - loss: 0.4300 - val_accuracy: 0.8171 - val_loss: 0.4837 -
learning_rate: 1.0000e-04
Epoch 8/50
139/139          1s 9ms/step
- val_f1: 0.8889 - val_precision: 0.8933 - val_recall: 0.8846
222/222          20s 88ms/step -
accuracy: 0.8703 - loss: 0.4029 - val_accuracy: 0.8163 - val_loss: 0.4896 -
learning_rate: 1.0000e-04
Epoch 9/50
139/139          2s 13ms/step
- val_f1: 0.8859 - val_precision: 0.8908 - val_recall: 0.8811
222/222          19s 87ms/step -

```

accuracy: 0.8789 - loss: 0.3717 - val_accuracy: 0.8168 - val_loss: 0.5024 -
learning_rate: 1.0000e-04
Epoch 10/50
139/139 1s 8ms/step
- val_f1: 0.8905 - val_precision: 0.8856 - val_recall: 0.8955
222/222 21s 89ms/step -
accuracy: 0.8931 - loss: 0.3517 - val_accuracy: 0.8208 - val_loss: 0.5060 -
learning_rate: 1.0000e-04
139/139 1s 9ms/step
Accuracy: 0.8318
Precision: 0.8948
Recall: 0.8840
F1-score: 0.8894
ROC AUC: 0.8648
Confusion Matrix:
[[688 352]
 [393 2995]]





Epoch 1/50

7/7 0s 28ms/step

- val_f1: 0.5520 - val_precision: 0.6703 - val_recall: 0.4692

11/11 4s 191ms/step -

accuracy: 0.5199 - loss: 3.4466 - val_accuracy: 0.5503 - val_loss: 3.3734 -

learning_rate: 1.0000e-04

Epoch 2/50

7/7 0s 23ms/step

- val_f1: 0.4894 - val_precision: 0.7931 - val_recall: 0.3538

11/11 2s 208ms/step -

accuracy: 0.5408 - loss: 3.3440 - val_accuracy: 0.5740 - val_loss: 3.2736 -

learning_rate: 1.0000e-04

Epoch 3/50

7/7 0s 23ms/step

- val_f1: 0.3636 - val_precision: 0.8571 - val_recall: 0.2308

11/11 3s 211ms/step -

accuracy: 0.5131 - loss: 3.2449 - val_accuracy: 0.4911 - val_loss: 3.1774 -

learning_rate: 1.0000e-04

Epoch 4/50

7/7 0s 17ms/step

```

- val_f1: 0.3860 - val_precision: 0.8049 - val_recall: 0.2538
11/11          2s 137ms/step -
accuracy: 0.5551 - loss: 3.1478 - val_accuracy: 0.5089 - val_loss: 3.0846 -
learning_rate: 1.0000e-04
Epoch 5/50
7/7           0s 14ms/step
- val_f1: 0.3473 - val_precision: 0.7838 - val_recall: 0.2231
11/11          3s 145ms/step -
accuracy: 0.5500 - loss: 3.0560 - val_accuracy: 0.4970 - val_loss: 2.9955 -
learning_rate: 1.0000e-04
Epoch 6/50
7/7           0s 14ms/step
- val_f1: 0.3953 - val_precision: 0.8095 - val_recall: 0.2615
11/11          2s 150ms/step -
accuracy: 0.5314 - loss: 2.9663 - val_accuracy: 0.5089 - val_loss: 2.9097 -
learning_rate: 1.0000e-04
Epoch 7/50
7/7           0s 15ms/step
- val_f1: 0.4444 - val_precision: 0.8000 - val_recall: 0.3077
11/11          2s 145ms/step -
accuracy: 0.5491 - loss: 2.8805 - val_accuracy: 0.5503 - val_loss: 2.8271 -
learning_rate: 1.0000e-04
Epoch 8/50
7/7           0s 31ms/step
- val_f1: 0.4731 - val_precision: 0.7857 - val_recall: 0.3385
11/11          4s 245ms/step -
accuracy: 0.5716 - loss: 2.7985 - val_accuracy: 0.5740 - val_loss: 2.7477 -
learning_rate: 1.0000e-04
Epoch 9/50
7/7           0s 14ms/step
- val_f1: 0.5026 - val_precision: 0.7869 - val_recall: 0.3692
11/11          4s 152ms/step -
accuracy: 0.5852 - loss: 2.7197 - val_accuracy: 0.5680 - val_loss: 2.6717 -
learning_rate: 1.0000e-04
Epoch 10/50
7/7           0s 15ms/step
- val_f1: 0.5258 - val_precision: 0.7969 - val_recall: 0.3923
11/11          2s 134ms/step -
accuracy: 0.6081 - loss: 2.6439 - val_accuracy: 0.5740 - val_loss: 2.5987 -
learning_rate: 1.0000e-04
Epoch 11/50
7/7           0s 14ms/step
- val_f1: 0.5381 - val_precision: 0.7910 - val_recall: 0.4077
11/11          1s 135ms/step -
accuracy: 0.5931 - loss: 2.5709 - val_accuracy: 0.5799 - val_loss: 2.5284 -
learning_rate: 1.0000e-04
Epoch 12/50
7/7           0s 15ms/step

```

```

- val_f1: 0.5381 - val_precision: 0.7910 - val_recall: 0.4077
11/11          3s 142ms/step -
accuracy: 0.6228 - loss: 2.5012 - val_accuracy: 0.5799 - val_loss: 2.4610 -
learning_rate: 1.0000e-04
Epoch 13/50
7/7           0s 14ms/step
- val_f1: 0.5528 - val_precision: 0.7971 - val_recall: 0.4231
11/11          2s 141ms/step -
accuracy: 0.6171 - loss: 2.4339 - val_accuracy: 0.5799 - val_loss: 2.3962 -
learning_rate: 1.0000e-04
Epoch 14/50
7/7           0s 23ms/step
- val_f1: 0.5528 - val_precision: 0.7971 - val_recall: 0.4231
11/11          2s 153ms/step -
accuracy: 0.6070 - loss: 2.3687 - val_accuracy: 0.5799 - val_loss: 2.3341 -
learning_rate: 1.0000e-04
Epoch 15/50
7/7           0s 24ms/step
- val_f1: 0.5672 - val_precision: 0.8028 - val_recall: 0.4385
11/11          2s 208ms/step -
accuracy: 0.6136 - loss: 2.3062 - val_accuracy: 0.5799 - val_loss: 2.2742 -
learning_rate: 1.0000e-04
Epoch 16/50
7/7           0s 24ms/step
- val_f1: 0.5854 - val_precision: 0.8000 - val_recall: 0.4615
11/11          3s 227ms/step -
accuracy: 0.6062 - loss: 2.2483 - val_accuracy: 0.5976 - val_loss: 2.2164 -
learning_rate: 1.0000e-04
Epoch 17/50
7/7           0s 14ms/step
- val_f1: 0.5813 - val_precision: 0.8082 - val_recall: 0.4538
11/11          2s 145ms/step -
accuracy: 0.6588 - loss: 2.1895 - val_accuracy: 0.5976 - val_loss: 2.1613 -
learning_rate: 1.0000e-04
Epoch 18/50
7/7           0s 21ms/step
- val_f1: 0.5854 - val_precision: 0.8000 - val_recall: 0.4615
11/11          3s 154ms/step -
accuracy: 0.6299 - loss: 2.1349 - val_accuracy: 0.5976 - val_loss: 2.1081 -
learning_rate: 1.0000e-04
Epoch 19/50
7/7           0s 16ms/step
- val_f1: 0.5784 - val_precision: 0.7973 - val_recall: 0.4538
11/11          2s 136ms/step -
accuracy: 0.6777 - loss: 2.0810 - val_accuracy: 0.5976 - val_loss: 2.0572 -
learning_rate: 1.0000e-04
Epoch 20/50
7/7           0s 14ms/step

```

```

- val_f1: 0.5784 - val_precision: 0.7973 - val_recall: 0.4538
11/11          3s 137ms/step -
accuracy: 0.6405 - loss: 2.0301 - val_accuracy: 0.5976 - val_loss: 2.0084 -
learning_rate: 1.0000e-04
Epoch 21/50
7/7           0s 26ms/step
- val_f1: 0.5784 - val_precision: 0.7973 - val_recall: 0.4538
11/11          2s 188ms/step -
accuracy: 0.6611 - loss: 1.9806 - val_accuracy: 0.5976 - val_loss: 1.9612 -
learning_rate: 1.0000e-04
Epoch 22/50
7/7           0s 24ms/step
- val_f1: 0.5784 - val_precision: 0.7973 - val_recall: 0.4538
11/11          2s 220ms/step -
accuracy: 0.6648 - loss: 1.9326 - val_accuracy: 0.5976 - val_loss: 1.9161 -
learning_rate: 1.0000e-04
Epoch 23/50
7/7           0s 14ms/step
- val_f1: 0.5962 - val_precision: 0.7949 - val_recall: 0.4769
11/11          2s 170ms/step -
accuracy: 0.6433 - loss: 1.8880 - val_accuracy: 0.5917 - val_loss: 1.8723 -
learning_rate: 1.0000e-04
Epoch 24/50
7/7           0s 16ms/step
- val_f1: 0.6029 - val_precision: 0.7975 - val_recall: 0.4846
11/11          2s 134ms/step -
accuracy: 0.6625 - loss: 1.8476 - val_accuracy: 0.5917 - val_loss: 1.8307 -
learning_rate: 1.0000e-04
Epoch 25/50
7/7           0s 14ms/step
- val_f1: 0.6029 - val_precision: 0.7975 - val_recall: 0.4846
11/11          3s 145ms/step -
accuracy: 0.6726 - loss: 1.8032 - val_accuracy: 0.5976 - val_loss: 1.7905 -
learning_rate: 1.0000e-04
Epoch 26/50
7/7           0s 15ms/step
- val_f1: 0.5894 - val_precision: 0.7922 - val_recall: 0.4692
11/11          2s 139ms/step -
accuracy: 0.6613 - loss: 1.7615 - val_accuracy: 0.5917 - val_loss: 1.7524 -
learning_rate: 1.0000e-04
Epoch 27/50
7/7           0s 14ms/step
- val_f1: 0.5894 - val_precision: 0.7922 - val_recall: 0.4692
11/11          2s 138ms/step -
accuracy: 0.6430 - loss: 1.7235 - val_accuracy: 0.5917 - val_loss: 1.7154 -
learning_rate: 1.0000e-04
Epoch 28/50
7/7           0s 24ms/step

```

```

- val_f1: 0.6226 - val_precision: 0.8049 - val_recall: 0.5077
11/11          2s 198ms/step -
accuracy: 0.6493 - loss: 1.6853 - val_accuracy: 0.6036 - val_loss: 1.6796 -
learning_rate: 1.0000e-04
Epoch 29/50
7/7           0s 21ms/step
- val_f1: 0.6095 - val_precision: 0.8000 - val_recall: 0.4923
11/11          3s 222ms/step -
accuracy: 0.6859 - loss: 1.6486 - val_accuracy: 0.6036 - val_loss: 1.6457 -
learning_rate: 1.0000e-04
Epoch 30/50
7/7           0s 15ms/step
- val_f1: 0.6226 - val_precision: 0.8049 - val_recall: 0.5077
11/11          2s 143ms/step -
accuracy: 0.6709 - loss: 1.6135 - val_accuracy: 0.6036 - val_loss: 1.6128 -
learning_rate: 1.0000e-04
Epoch 31/50
7/7           0s 15ms/step
- val_f1: 0.6161 - val_precision: 0.8025 - val_recall: 0.5000
11/11          2s 139ms/step -
accuracy: 0.6866 - loss: 1.5772 - val_accuracy: 0.6036 - val_loss: 1.5815 -
learning_rate: 1.0000e-04
Epoch 32/50
7/7           0s 14ms/step
- val_f1: 0.6355 - val_precision: 0.8095 - val_recall: 0.5231
11/11          3s 138ms/step -
accuracy: 0.6750 - loss: 1.5444 - val_accuracy: 0.6213 - val_loss: 1.5508 -
learning_rate: 1.0000e-04
Epoch 33/50
7/7           0s 15ms/step
- val_f1: 0.6544 - val_precision: 0.8161 - val_recall: 0.5462
11/11          3s 137ms/step -
accuracy: 0.6792 - loss: 1.5118 - val_accuracy: 0.6154 - val_loss: 1.5212 -
learning_rate: 1.0000e-04
Epoch 34/50
7/7           0s 23ms/step
- val_f1: 0.6481 - val_precision: 0.8140 - val_recall: 0.5385
11/11          2s 177ms/step -
accuracy: 0.6978 - loss: 1.4777 - val_accuracy: 0.6154 - val_loss: 1.4930 -
learning_rate: 1.0000e-04
Epoch 35/50
7/7           0s 24ms/step
- val_f1: 0.6574 - val_precision: 0.8256 - val_recall: 0.5462
11/11          2s 211ms/step -
accuracy: 0.7131 - loss: 1.4434 - val_accuracy: 0.6213 - val_loss: 1.4653 -
learning_rate: 1.0000e-04
Epoch 36/50
7/7           0s 14ms/step

```

```

- val_f1: 0.6758 - val_precision: 0.8315 - val_recall: 0.5692
11/11          2s 187ms/step -
accuracy: 0.7246 - loss: 1.4081 - val_accuracy: 0.6272 - val_loss: 1.4376 -
learning_rate: 1.0000e-04
Epoch 37/50
7/7           0s 16ms/step
- val_f1: 0.7168 - val_precision: 0.8438 - val_recall: 0.6231
11/11          2s 138ms/step -
accuracy: 0.7385 - loss: 1.3795 - val_accuracy: 0.6686 - val_loss: 1.4098 -
learning_rate: 1.0000e-04
Epoch 38/50
7/7           0s 14ms/step
- val_f1: 0.7225 - val_precision: 0.8454 - val_recall: 0.6308
11/11          3s 144ms/step -
accuracy: 0.7841 - loss: 1.3374 - val_accuracy: 0.6805 - val_loss: 1.3829 -
learning_rate: 1.0000e-04
Epoch 39/50
7/7           0s 14ms/step
- val_f1: 0.7225 - val_precision: 0.8454 - val_recall: 0.6308
11/11          3s 142ms/step -
accuracy: 0.8081 - loss: 1.2946 - val_accuracy: 0.6923 - val_loss: 1.3548 -
learning_rate: 1.0000e-04
Epoch 40/50
7/7           0s 14ms/step
- val_f1: 0.7336 - val_precision: 0.8485 - val_recall: 0.6462
11/11          2s 135ms/step -
accuracy: 0.8427 - loss: 1.2564 - val_accuracy: 0.6982 - val_loss: 1.3299 -
learning_rate: 1.0000e-04
Epoch 41/50
7/7           0s 25ms/step
- val_f1: 0.7731 - val_precision: 0.8519 - val_recall: 0.7077
11/11          2s 195ms/step -
accuracy: 0.8446 - loss: 1.2114 - val_accuracy: 0.7160 - val_loss: 1.2994 -
learning_rate: 1.0000e-04
Epoch 42/50
7/7           0s 23ms/step
- val_f1: 0.7884 - val_precision: 0.8559 - val_recall: 0.7308
11/11          3s 224ms/step -
accuracy: 0.8655 - loss: 1.1665 - val_accuracy: 0.7219 - val_loss: 1.2715 -
learning_rate: 1.0000e-04
Epoch 43/50
7/7           0s 17ms/step
- val_f1: 0.8130 - val_precision: 0.8621 - val_recall: 0.7692
11/11          2s 138ms/step -
accuracy: 0.8877 - loss: 1.1128 - val_accuracy: 0.7456 - val_loss: 1.2401 -
learning_rate: 1.0000e-04
Epoch 44/50
7/7           0s 15ms/step

```

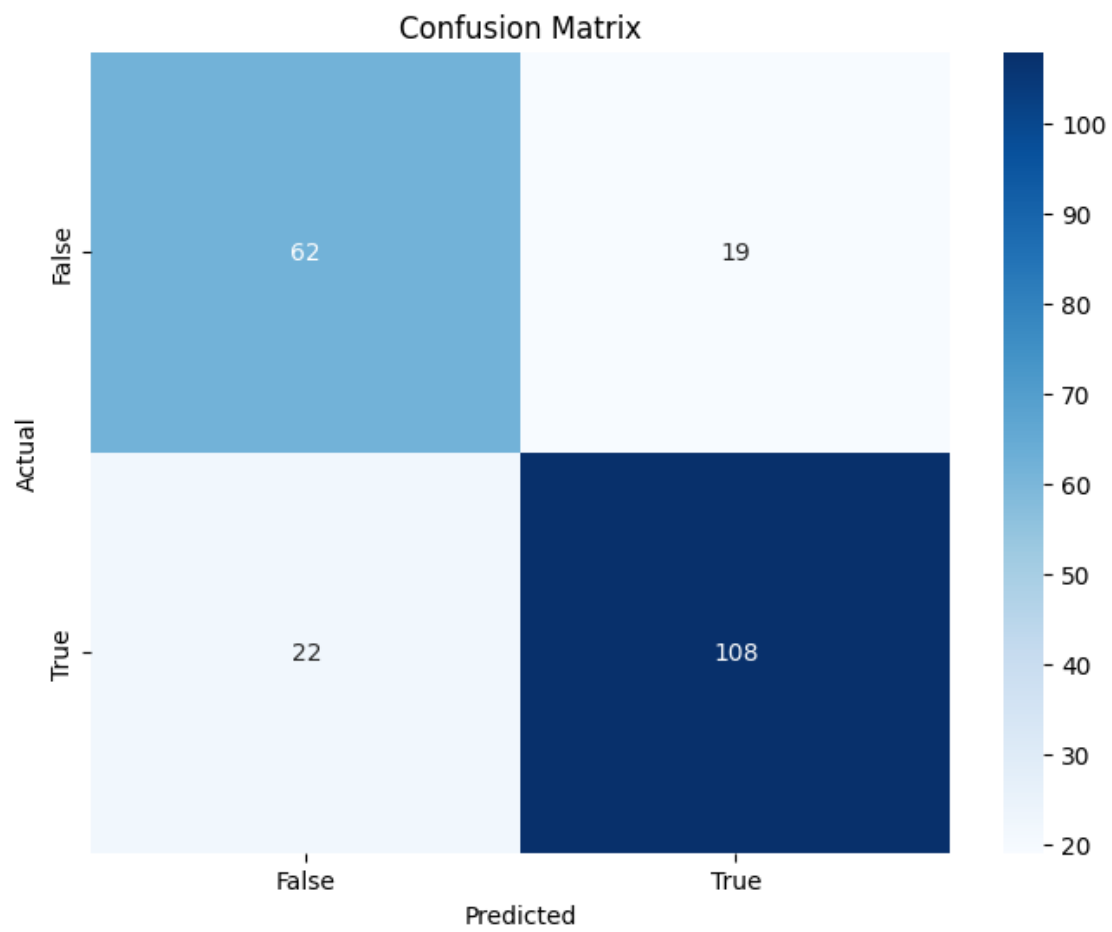


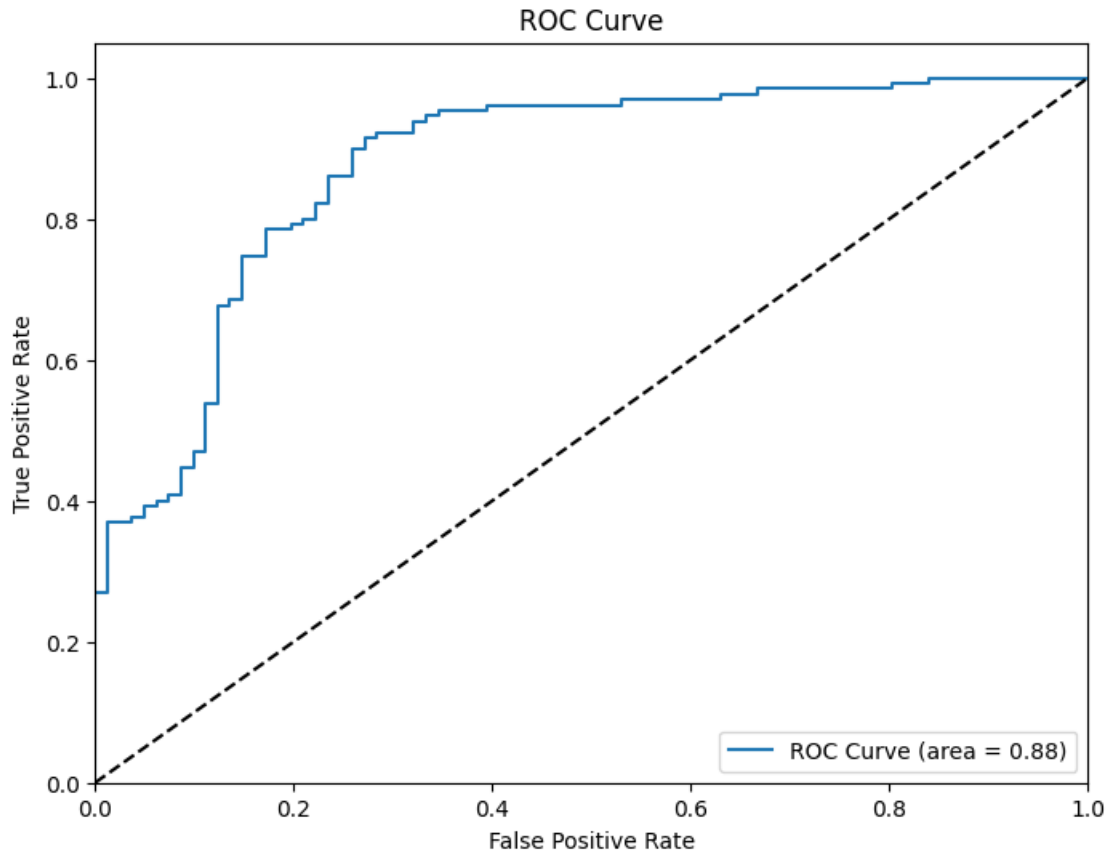
```

- val_f1: 0.8379 - val_precision: 0.8618 - val_recall: 0.8154
11/11          3s 145ms/step -
accuracy: 0.9164 - loss: 1.0637 - val_accuracy: 0.7574 - val_loss: 1.2110 -
learning_rate: 1.0000e-04
Epoch 45/50
7/7           0s 16ms/step
- val_f1: 0.8392 - val_precision: 0.8560 - val_recall: 0.8231
11/11          2s 143ms/step -
accuracy: 0.9272 - loss: 1.0242 - val_accuracy: 0.7574 - val_loss: 1.1850 -
learning_rate: 1.0000e-04
Epoch 46/50
7/7           0s 15ms/step
- val_f1: 0.8300 - val_precision: 0.8537 - val_recall: 0.8077
11/11          2s 139ms/step -
accuracy: 0.9338 - loss: 0.9899 - val_accuracy: 0.7574 - val_loss: 1.1690 -
learning_rate: 1.0000e-04
Epoch 47/50
7/7           0s 16ms/step
- val_f1: 0.8392 - val_precision: 0.8560 - val_recall: 0.8231
11/11          3s 138ms/step -
accuracy: 0.9411 - loss: 0.9439 - val_accuracy: 0.7633 - val_loss: 1.1432 -
learning_rate: 1.0000e-04
Epoch 48/50
7/7           0s 26ms/step
- val_f1: 0.8346 - val_precision: 0.8548 - val_recall: 0.8154
11/11          3s 222ms/step -
accuracy: 0.9452 - loss: 0.9129 - val_accuracy: 0.7633 - val_loss: 1.1298 -
learning_rate: 1.0000e-04
Epoch 49/50
7/7           0s 15ms/step
- val_f1: 0.8359 - val_precision: 0.8492 - val_recall: 0.8231
11/11          2s 179ms/step -
accuracy: 0.9453 - loss: 0.8968 - val_accuracy: 0.7692 - val_loss: 1.1144 -
learning_rate: 1.0000e-04
Epoch 50/50
7/7           0s 17ms/step
- val_f1: 0.8405 - val_precision: 0.8504 - val_recall: 0.8308
11/11          2s 137ms/step -
accuracy: 0.9462 - loss: 0.8765 - val_accuracy: 0.7811 - val_loss: 1.1007 -
learning_rate: 1.0000e-04
7/7           0s 14ms/step
Accuracy: 0.8057
Precision: 0.8504
Recall: 0.8308
F1-score: 0.8405
ROC AUC: 0.8760
Confusion Matrix:
[[ 62  19]

```

```
[ 22 108]
```





1.8.2 Model Performance over Epochs

The training history including F1 Score and loss for both training and validation sets is plotted for each dataset.

```
[14]: def plot_training_history(history, dataset_name):
    # Plot training & validation F1 score values
    plt.figure(figsize=(12, 6))
    plt.suptitle(f'{dataset_name} - Model Performance Over Epochs',
    ↪fontsize=16, y=1.02) # y parameter for spacing

    plt.subplot(1, 2, 1)
    if 'val_f1_score' in history.history:
        plt.plot(history.history['val_f1_score'])
        plt.title('Model F1 Score')
        plt.ylabel('F1 Score')
    else:
        plt.plot(history.history['accuracy'])
        plt.title('Model Accuracy')
        plt.ylabel('Accuracy')
```

```

plt.plot(history.history['val_accuracy'])
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')

# Plot training & validation loss values
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')

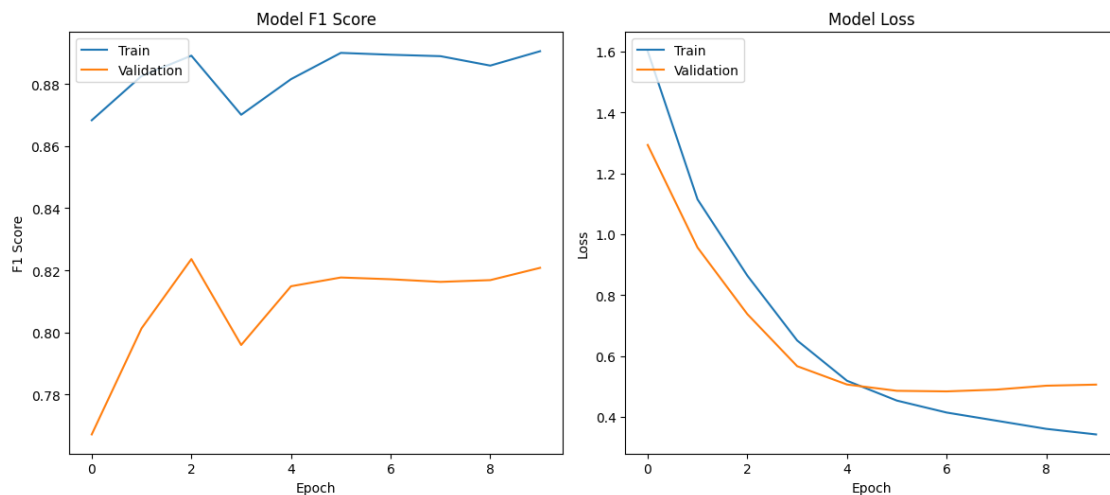
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()

# Plot training history for GossipCop
plot_training_history(history_gossipcop, "GossipCop")

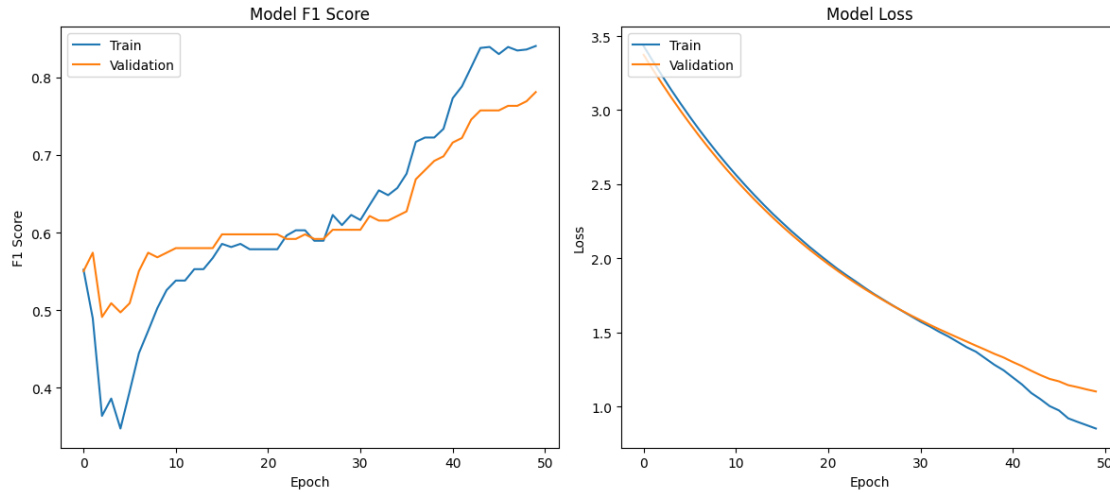
# Plot training history for PolitiFact
plot_training_history(history_politifact, "PolitiFact")

```

GossipCop - Model Performance Over Epochs



PolitiFact - Model Performance Over Epochs



F1 score graphs show an increase over the epochs. It means the models are learning and improving their ability to balance precision and recall.

The loss graphs show a steady decrease in both training and validation loss at the beginning, that stabilizes towards the later epochs. This means the models are converging and learning effectively without overfitting.

1.8.3 Comparison of my results with baseline Shu et al. [7]

Shu et al. Baseline Results:

GossipCop

Accuracy: 0.723
Precision: 0.751
Recall: 0.701
F1: 0.725

PolitiFact

Accuracy: 0.629
Precision: 0.807
Recall: 0.456
F1: 0.583

My Model Results:

GossipCop

Accuracy: 0.8318
Precision: 0.8948
Recall: 0.8840
F1: 0.8894

PolitiFact

Accuracy: 0.8057
Precision: 0.8504
Recall: 0.8308
F1: 0.8405

My models significantly outperformed Shu et al.'s CNN results in all metrics for both datasets.

1.9 Implement 5-Fold Cross-Validation

I will randomly split the datasets into five parts and conduct 5-fold cross-validation to obtain robust results, as Bian et al.[5] did. Cross-validation is a statistical method used to estimate the performance of machine learning models. As Browne et al.[15] explain: “In its simplest form, the leaving one out at a time method, this involves partitioning a sample of size N into a calibration sample of size N-1 and a validation sample of size 1 and repeating the process N times. An average of the N cross-validation index values is then used.” This method involves splitting the data into a number of subsets (folds), training the model on some subsets while testing it on the remaining subset, and repeating this process several times. The performance metrics are then averaged over all iterations to provide a more robust evaluation.

1.9.1 Cross-Validation Function

I could not use scikeras for k-fold cross validation due to compatibility issues with existing models, so I implemented k-fold cross validation manually.

```
[15]: # Function to perform cross-validation
def run_kfold_cross_validation(X_data, y_data, create_model_func, dataset_name,
    n_splits=5):
    X = np.array(X_data)
    y = np.array(y_data)

    kf = KFold(n_splits=n_splits, shuffle=True, random_state=42)
    fold_no = 1

    # Initialize lists to store fold-wise performance metrics
    f1_scores, precision_scores, recall_scores = [], [], []

    for train_index, test_index in kf.split(X):
        X_train, X_test = X[train_index], X[test_index]
        y_train, y_test = y[train_index], y[test_index]

        # Clearing the TensorFlow backend to prevent data leakage
        clear_session()

        # Create a new instance of the CNN model
        model = create_model_func(input_length)

        # Train the model
        print(f'Training for fold {fold_no} on {dataset_name}...')
```

```

        model.fit(X_train, y_train, epochs=50, batch_size=64,
↪validation_data=(X_test, y_test))

        # Evaluate the model
        y_pred = model.predict(X_test)
        y_pred = (y_pred > 0.5).astype(int)

        # Calculate metrics
        f1 = f1_score(y_test, y_pred)
        precision = precision_score(y_test, y_pred)
        recall = recall_score(y_test, y_pred)

        f1_scores.append(f1)
        precision_scores.append(precision)
        recall_scores.append(recall)

        print(f'Fold {fold_no} - Precision: {precision:.4f}, Recall: {recall:.
↪4f}, F1 Score: {f1:.4f}')

        fold_no += 1

    # Return the average scores
    print(f'Average Precision: {np.mean(precision_scores):.4f}')
    print(f'Average Recall: {np.mean(recall_scores):.4f}')
    print(f'Average F1 Score: {np.mean(f1_scores):.4f}')

```

1.9.2 Cross-Validation Scores

```

[16]: # Run cross-validation
run_kfold_cross_validation(X_gossipcop, y_gossipcop,
↪create_cnn_model_gossipcop, "GossipCop")
run_kfold_cross_validation(X_politifact, y_politifact,
↪create_cnn_model_politifact, "PolitiFact")

```

Training for fold 1 on GossipCop...

Epoch 1/50

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:

UserWarning: Argument `input_length` is deprecated. Just remove it.

warnings.warn(

277/277 26s 84ms/step -

accuracy: 0.7266 - loss: 1.6738 - val_accuracy: 0.7651 - val_loss: 1.0707

Epoch 2/50

277/277 41s 84ms/step -

accuracy: 0.7560 - loss: 0.9933 - val_accuracy: 0.7651 - val_loss: 0.7552

Epoch 3/50

277/277 38s 73ms/step -

accuracy: 0.7560 - loss: 0.7377 - val_accuracy: 0.7651 - val_loss: 0.6311
 Epoch 4/50
 277/277 23s 83ms/step -
 accuracy: 0.7560 - loss: 0.6369 - val_accuracy: 0.7651 - val_loss: 0.5842
 Epoch 5/50
 277/277 39s 75ms/step -
 accuracy: 0.7560 - loss: 0.5990 - val_accuracy: 0.7651 - val_loss: 0.5660
 Epoch 6/50
 277/277 41s 74ms/step -
 accuracy: 0.7560 - loss: 0.5845 - val_accuracy: 0.7651 - val_loss: 0.5587
 Epoch 7/50
 277/277 41s 74ms/step -
 accuracy: 0.7560 - loss: 0.5756 - val_accuracy: 0.7651 - val_loss: 0.5547
 Epoch 8/50
 277/277 43s 83ms/step -
 accuracy: 0.7560 - loss: 0.5718 - val_accuracy: 0.7651 - val_loss: 0.5525
 Epoch 9/50
 277/277 38s 73ms/step -
 accuracy: 0.7560 - loss: 0.5728 - val_accuracy: 0.7651 - val_loss: 0.5487
 Epoch 10/50
 277/277 22s 78ms/step -
 accuracy: 0.7560 - loss: 0.5636 - val_accuracy: 0.7651 - val_loss: 0.5128
 Epoch 11/50
 277/277 42s 81ms/step -
 accuracy: 0.7635 - loss: 0.5077 - val_accuracy: 0.7949 - val_loss: 0.4691
 Epoch 12/50
 277/277 39s 74ms/step -
 accuracy: 0.7997 - loss: 0.4500 - val_accuracy: 0.8071 - val_loss: 0.4605
 Epoch 13/50
 277/277 41s 74ms/step -
 accuracy: 0.8238 - loss: 0.4204 - val_accuracy: 0.8119 - val_loss: 0.4585
 Epoch 14/50
 277/277 23s 81ms/step -
 accuracy: 0.8396 - loss: 0.4005 - val_accuracy: 0.8132 - val_loss: 0.4558
 Epoch 15/50
 277/277 39s 75ms/step -
 accuracy: 0.8474 - loss: 0.3851 - val_accuracy: 0.8171 - val_loss: 0.4543
 Epoch 16/50
 277/277 41s 75ms/step -
 accuracy: 0.8584 - loss: 0.3723 - val_accuracy: 0.8209 - val_loss: 0.4517
 Epoch 17/50
 277/277 22s 79ms/step -
 accuracy: 0.8674 - loss: 0.3565 - val_accuracy: 0.8238 - val_loss: 0.4459
 Epoch 18/50
 277/277 41s 78ms/step -
 accuracy: 0.8798 - loss: 0.3394 - val_accuracy: 0.8290 - val_loss: 0.4453
 Epoch 19/50
 277/277 40s 75ms/step -

accuracy: 0.8845 - loss: 0.3280 - val_accuracy: 0.8304 - val_loss: 0.4481
 Epoch 20/50
 277/277 43s 82ms/step -
 accuracy: 0.8872 - loss: 0.3182 - val_accuracy: 0.8299 - val_loss: 0.4523
 Epoch 21/50
 277/277 22s 78ms/step -
 accuracy: 0.8909 - loss: 0.3118 - val_accuracy: 0.8304 - val_loss: 0.4529
 Epoch 22/50
 277/277 42s 82ms/step -
 accuracy: 0.9004 - loss: 0.2969 - val_accuracy: 0.8324 - val_loss: 0.4599
 Epoch 23/50
 277/277 39s 75ms/step -
 accuracy: 0.9034 - loss: 0.2929 - val_accuracy: 0.8338 - val_loss: 0.4656
 Epoch 24/50
 277/277 45s 89ms/step -
 accuracy: 0.9089 - loss: 0.2806 - val_accuracy: 0.8324 - val_loss: 0.4747
 Epoch 25/50
 277/277 37s 75ms/step -
 accuracy: 0.9109 - loss: 0.2756 - val_accuracy: 0.8308 - val_loss: 0.4824
 Epoch 26/50
 277/277 41s 74ms/step -
 accuracy: 0.9161 - loss: 0.2683 - val_accuracy: 0.8315 - val_loss: 0.4905
 Epoch 27/50
 277/277 22s 80ms/step -
 accuracy: 0.9216 - loss: 0.2628 - val_accuracy: 0.8297 - val_loss: 0.4984
 Epoch 28/50
 277/277 41s 79ms/step -
 accuracy: 0.9216 - loss: 0.2533 - val_accuracy: 0.8299 - val_loss: 0.5090
 Epoch 29/50
 277/277 23s 84ms/step -
 accuracy: 0.9263 - loss: 0.2475 - val_accuracy: 0.8308 - val_loss: 0.5177
 Epoch 30/50
 277/277 40s 80ms/step -
 accuracy: 0.9290 - loss: 0.2375 - val_accuracy: 0.8306 - val_loss: 0.5291
 Epoch 31/50
 277/277 40s 75ms/step -
 accuracy: 0.9355 - loss: 0.2351 - val_accuracy: 0.8315 - val_loss: 0.5367
 Epoch 32/50
 277/277 41s 74ms/step -
 accuracy: 0.9368 - loss: 0.2281 - val_accuracy: 0.8324 - val_loss: 0.5467
 Epoch 33/50
 277/277 43s 82ms/step -
 accuracy: 0.9399 - loss: 0.2172 - val_accuracy: 0.8318 - val_loss: 0.5618
 Epoch 34/50
 277/277 39s 74ms/step -
 accuracy: 0.9430 - loss: 0.2155 - val_accuracy: 0.8327 - val_loss: 0.5686
 Epoch 35/50
 277/277 41s 73ms/step -

accuracy: 0.9445 - loss: 0.2064 - val_accuracy: 0.8322 - val_loss: 0.5795
 Epoch 36/50
 277/277 22s 79ms/step -
 accuracy: 0.9484 - loss: 0.2031 - val_accuracy: 0.8308 - val_loss: 0.5969
 Epoch 37/50
 277/277 21s 75ms/step -
 accuracy: 0.9521 - loss: 0.1953 - val_accuracy: 0.8315 - val_loss: 0.5999
 Epoch 38/50
 277/277 43s 83ms/step -
 accuracy: 0.9520 - loss: 0.1918 - val_accuracy: 0.8311 - val_loss: 0.6080
 Epoch 39/50
 277/277 40s 80ms/step -
 accuracy: 0.9543 - loss: 0.1853 - val_accuracy: 0.8295 - val_loss: 0.6240
 Epoch 40/50
 277/277 20s 73ms/step -
 accuracy: 0.9564 - loss: 0.1782 - val_accuracy: 0.8277 - val_loss: 0.6316
 Epoch 41/50
 277/277 21s 76ms/step -
 accuracy: 0.9594 - loss: 0.1726 - val_accuracy: 0.8290 - val_loss: 0.6525
 Epoch 42/50
 277/277 40s 74ms/step -
 accuracy: 0.9607 - loss: 0.1723 - val_accuracy: 0.8272 - val_loss: 0.6576
 Epoch 43/50
 277/277 42s 79ms/step -
 accuracy: 0.9628 - loss: 0.1640 - val_accuracy: 0.8238 - val_loss: 0.6661
 Epoch 44/50
 277/277 39s 73ms/step -
 accuracy: 0.9633 - loss: 0.1610 - val_accuracy: 0.8241 - val_loss: 0.6811
 Epoch 45/50
 277/277 22s 79ms/step -
 accuracy: 0.9661 - loss: 0.1555 - val_accuracy: 0.8229 - val_loss: 0.6887
 Epoch 46/50
 277/277 40s 77ms/step -
 accuracy: 0.9695 - loss: 0.1486 - val_accuracy: 0.8254 - val_loss: 0.7002
 Epoch 47/50
 277/277 42s 79ms/step -
 accuracy: 0.9696 - loss: 0.1457 - val_accuracy: 0.8243 - val_loss: 0.7157
 Epoch 48/50
 277/277 40s 74ms/step -
 accuracy: 0.9687 - loss: 0.1443 - val_accuracy: 0.8254 - val_loss: 0.7186
 Epoch 49/50
 277/277 41s 74ms/step -
 accuracy: 0.9716 - loss: 0.1415 - val_accuracy: 0.8245 - val_loss: 0.7283
 Epoch 50/50
 277/277 22s 80ms/step -
 accuracy: 0.9708 - loss: 0.1376 - val_accuracy: 0.8245 - val_loss: 0.7521
 139/139 1s 9ms/step
 Fold 1 - Precision: 0.8644, Recall: 0.9141, F1 Score: 0.8885

Training for fold 2 on GossipCop...

Epoch 1/50

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:

UserWarning: Argument `input_length` is deprecated. Just remove it.

warnings.warn(

277/277 23s 75ms/step -

accuracy: 0.6808 - loss: 1.6950 - val_accuracy: 0.7565 - val_loss: 1.0686

Epoch 2/50

277/277 43s 82ms/step -

accuracy: 0.7576 - loss: 0.9791 - val_accuracy: 0.7565 - val_loss: 0.7532

Epoch 3/50

277/277 38s 73ms/step -

accuracy: 0.7576 - loss: 0.7269 - val_accuracy: 0.7565 - val_loss: 0.6334

Epoch 4/50

277/277 22s 80ms/step -

accuracy: 0.7576 - loss: 0.6298 - val_accuracy: 0.7565 - val_loss: 0.5893

Epoch 5/50

277/277 40s 75ms/step -

accuracy: 0.7576 - loss: 0.5950 - val_accuracy: 0.7565 - val_loss: 0.5732

Epoch 6/50

277/277 41s 74ms/step -

accuracy: 0.7576 - loss: 0.5815 - val_accuracy: 0.7565 - val_loss: 0.5667

Epoch 7/50

277/277 41s 74ms/step -

accuracy: 0.7576 - loss: 0.5756 - val_accuracy: 0.7565 - val_loss: 0.5639

Epoch 8/50

277/277 41s 74ms/step -

accuracy: 0.7576 - loss: 0.5712 - val_accuracy: 0.7565 - val_loss: 0.5621

Epoch 9/50

277/277 22s 80ms/step -

accuracy: 0.7576 - loss: 0.5701 - val_accuracy: 0.7565 - val_loss: 0.5611

Epoch 10/50

277/277 41s 81ms/step -

accuracy: 0.7576 - loss: 0.5717 - val_accuracy: 0.7565 - val_loss: 0.5602

Epoch 11/50

277/277 43s 88ms/step -

accuracy: 0.7576 - loss: 0.5672 - val_accuracy: 0.7565 - val_loss: 0.5596

Epoch 12/50

277/277 37s 74ms/step -

accuracy: 0.7576 - loss: 0.5661 - val_accuracy: 0.7565 - val_loss: 0.5592

Epoch 13/50

277/277 22s 80ms/step -

accuracy: 0.7576 - loss: 0.5683 - val_accuracy: 0.7565 - val_loss: 0.5586

Epoch 14/50

277/277 40s 76ms/step -

accuracy: 0.7576 - loss: 0.5661 - val_accuracy: 0.7565 - val_loss: 0.5583

Epoch 15/50

277/277 21s 77ms/step -
accuracy: 0.7576 - loss: 0.5662 - val_accuracy: 0.7565 - val_loss: 0.5579
Epoch 16/50

277/277 44s 87ms/step -
accuracy: 0.7576 - loss: 0.5670 - val_accuracy: 0.7565 - val_loss: 0.5574
Epoch 17/50

277/277 37s 74ms/step -
accuracy: 0.7576 - loss: 0.5665 - val_accuracy: 0.7565 - val_loss: 0.5561
Epoch 18/50

277/277 41s 75ms/step -
accuracy: 0.7576 - loss: 0.5619 - val_accuracy: 0.7565 - val_loss: 0.5213
Epoch 19/50

277/277 22s 80ms/step -
accuracy: 0.7576 - loss: 0.5053 - val_accuracy: 0.7565 - val_loss: 0.4764
Epoch 20/50

277/277 41s 82ms/step -
accuracy: 0.7636 - loss: 0.4570 - val_accuracy: 0.7565 - val_loss: 0.4651
Epoch 21/50

277/277 23s 81ms/step -
accuracy: 0.7916 - loss: 0.4346 - val_accuracy: 0.7565 - val_loss: 0.4602
Epoch 22/50

277/277 41s 80ms/step -
accuracy: 0.7908 - loss: 0.4222 - val_accuracy: 0.8130 - val_loss: 0.4570
Epoch 23/50

277/277 20s 73ms/step -
accuracy: 0.7993 - loss: 0.4103 - val_accuracy: 0.8187 - val_loss: 0.4547
Epoch 24/50

277/277 21s 76ms/step -
accuracy: 0.8050 - loss: 0.3987 - val_accuracy: 0.8198 - val_loss: 0.4538
Epoch 25/50

277/277 40s 74ms/step -
accuracy: 0.8196 - loss: 0.3871 - val_accuracy: 0.8214 - val_loss: 0.4534
Epoch 26/50

277/277 22s 80ms/step -
accuracy: 0.8300 - loss: 0.3747 - val_accuracy: 0.8263 - val_loss: 0.4513
Epoch 27/50

277/277 39s 73ms/step -
accuracy: 0.8361 - loss: 0.3655 - val_accuracy: 0.8275 - val_loss: 0.4518
Epoch 28/50

277/277 22s 80ms/step -
accuracy: 0.8388 - loss: 0.3544 - val_accuracy: 0.8308 - val_loss: 0.4507
Epoch 29/50

277/277 40s 78ms/step -
accuracy: 0.8411 - loss: 0.3475 - val_accuracy: 0.8313 - val_loss: 0.4516
Epoch 30/50

277/277 40s 75ms/step -
accuracy: 0.8461 - loss: 0.3377 - val_accuracy: 0.8311 - val_loss: 0.4554
Epoch 31/50

277/277 44s 88ms/step -
 accuracy: 0.8527 - loss: 0.3307 - val_accuracy: 0.8288 - val_loss: 0.4597
 Epoch 32/50
 277/277 37s 74ms/step -
 accuracy: 0.8528 - loss: 0.3282 - val_accuracy: 0.8299 - val_loss: 0.4613
 Epoch 33/50
 277/277 22s 80ms/step -
 accuracy: 0.8566 - loss: 0.3204 - val_accuracy: 0.8284 - val_loss: 0.4623
 Epoch 34/50
 277/277 41s 79ms/step -
 accuracy: 0.8587 - loss: 0.3135 - val_accuracy: 0.8270 - val_loss: 0.4696
 Epoch 35/50
 277/277 40s 74ms/step -
 accuracy: 0.8840 - loss: 0.3032 - val_accuracy: 0.8263 - val_loss: 0.4754
 Epoch 36/50
 277/277 41s 75ms/step -
 accuracy: 0.8881 - loss: 0.3043 - val_accuracy: 0.8268 - val_loss: 0.4816
 Epoch 37/50
 277/277 41s 74ms/step -
 accuracy: 0.8930 - loss: 0.2959 - val_accuracy: 0.8252 - val_loss: 0.4857
 Epoch 38/50
 277/277 41s 74ms/step -
 accuracy: 0.8942 - loss: 0.2944 - val_accuracy: 0.8257 - val_loss: 0.4946
 Epoch 39/50
 277/277 41s 73ms/step -
 accuracy: 0.9009 - loss: 0.2876 - val_accuracy: 0.8229 - val_loss: 0.5001
 Epoch 40/50
 277/277 24s 87ms/step -
 accuracy: 0.9052 - loss: 0.2817 - val_accuracy: 0.8254 - val_loss: 0.5106
 Epoch 41/50
 277/277 39s 81ms/step -
 accuracy: 0.9071 - loss: 0.2791 - val_accuracy: 0.8241 - val_loss: 0.5189
 Epoch 42/50
 277/277 39s 75ms/step -
 accuracy: 0.9100 - loss: 0.2690 - val_accuracy: 0.8245 - val_loss: 0.5270
 Epoch 43/50
 277/277 41s 74ms/step -
 accuracy: 0.9153 - loss: 0.2649 - val_accuracy: 0.8272 - val_loss: 0.5369
 Epoch 44/50
 277/277 41s 75ms/step -
 accuracy: 0.9171 - loss: 0.2640 - val_accuracy: 0.8254 - val_loss: 0.5494
 Epoch 45/50
 277/277 41s 76ms/step -
 accuracy: 0.9192 - loss: 0.2583 - val_accuracy: 0.8248 - val_loss: 0.5621
 Epoch 46/50
 277/277 22s 80ms/step -
 accuracy: 0.9248 - loss: 0.2481 - val_accuracy: 0.8257 - val_loss: 0.5703
 Epoch 47/50

```

277/277          40s 78ms/step -
accuracy: 0.9259 - loss: 0.2420 - val_accuracy: 0.8245 - val_loss: 0.5752
Epoch 48/50
277/277          40s 75ms/step -
accuracy: 0.9315 - loss: 0.2394 - val_accuracy: 0.8236 - val_loss: 0.5903
Epoch 49/50
277/277          43s 81ms/step -
accuracy: 0.9321 - loss: 0.2327 - val_accuracy: 0.8245 - val_loss: 0.5962
Epoch 50/50
277/277          23s 84ms/step -
accuracy: 0.9355 - loss: 0.2269 - val_accuracy: 0.8261 - val_loss: 0.6126
139/139          1s 9ms/step
Fold 2 - Precision: 0.8703, Recall: 0.9051, F1 Score: 0.8873
Training for fold 3 on GossipCop...
Epoch 1/50

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(

277/277          23s 74ms/step -
accuracy: 0.6654 - loss: 1.6911 - val_accuracy: 0.7545 - val_loss: 1.0713
Epoch 2/50
277/277          22s 80ms/step -
accuracy: 0.7603 - loss: 0.9795 - val_accuracy: 0.7545 - val_loss: 0.7577
Epoch 3/50
277/277          40s 78ms/step -
accuracy: 0.7603 - loss: 0.7262 - val_accuracy: 0.7545 - val_loss: 0.6378
Epoch 4/50
277/277          21s 76ms/step -
accuracy: 0.7603 - loss: 0.6296 - val_accuracy: 0.7545 - val_loss: 0.5931
Epoch 5/50
277/277          23s 81ms/step -
accuracy: 0.7603 - loss: 0.5911 - val_accuracy: 0.7545 - val_loss: 0.5763
Epoch 6/50
277/277          39s 75ms/step -
accuracy: 0.7603 - loss: 0.5769 - val_accuracy: 0.7545 - val_loss: 0.5694
Epoch 7/50
277/277          41s 74ms/step -
accuracy: 0.7603 - loss: 0.5744 - val_accuracy: 0.7545 - val_loss: 0.5664
Epoch 8/50
277/277          22s 80ms/step -
accuracy: 0.7603 - loss: 0.5694 - val_accuracy: 0.7545 - val_loss: 0.5645
Epoch 9/50
277/277          39s 75ms/step -
accuracy: 0.7603 - loss: 0.5674 - val_accuracy: 0.7545 - val_loss: 0.5632
Epoch 10/50
277/277          43s 81ms/step -
accuracy: 0.7603 - loss: 0.5668 - val_accuracy: 0.7545 - val_loss: 0.5622

```

Epoch 11/50
 277/277 22s 79ms/step -
 accuracy: 0.7603 - loss: 0.5646 - val_accuracy: 0.7545 - val_loss: 0.5615

Epoch 12/50
 277/277 41s 81ms/step -
 accuracy: 0.7603 - loss: 0.5647 - val_accuracy: 0.7545 - val_loss: 0.5610

Epoch 13/50
 277/277 21s 74ms/step -
 accuracy: 0.7603 - loss: 0.5638 - val_accuracy: 0.7545 - val_loss: 0.5608

Epoch 14/50
 277/277 41s 74ms/step -
 accuracy: 0.7603 - loss: 0.5634 - val_accuracy: 0.7545 - val_loss: 0.5604

Epoch 15/50
 277/277 20s 74ms/step -
 accuracy: 0.7603 - loss: 0.5619 - val_accuracy: 0.7545 - val_loss: 0.5601

Epoch 16/50
 277/277 25s 90ms/step -
 accuracy: 0.7603 - loss: 0.5622 - val_accuracy: 0.7545 - val_loss: 0.5602

Epoch 17/50
 277/277 37s 74ms/step -
 accuracy: 0.7603 - loss: 0.5621 - val_accuracy: 0.7545 - val_loss: 0.5598

Epoch 18/50
 277/277 41s 75ms/step -
 accuracy: 0.7603 - loss: 0.5586 - val_accuracy: 0.7545 - val_loss: 0.5596

Epoch 19/50
 277/277 41s 75ms/step -
 accuracy: 0.7603 - loss: 0.5605 - val_accuracy: 0.7545 - val_loss: 0.5594

Epoch 20/50
 277/277 43s 80ms/step -
 accuracy: 0.7603 - loss: 0.5587 - val_accuracy: 0.7545 - val_loss: 0.5592

Epoch 21/50
 277/277 39s 75ms/step -
 accuracy: 0.7603 - loss: 0.5608 - val_accuracy: 0.7545 - val_loss: 0.5591

Epoch 22/50
 277/277 21s 76ms/step -
 accuracy: 0.7603 - loss: 0.5608 - val_accuracy: 0.7545 - val_loss: 0.5591

Epoch 23/50
 277/277 40s 74ms/step -
 accuracy: 0.7603 - loss: 0.5596 - val_accuracy: 0.7545 - val_loss: 0.5589

Epoch 24/50
 277/277 22s 80ms/step -
 accuracy: 0.7603 - loss: 0.5580 - val_accuracy: 0.7545 - val_loss: 0.5588

Epoch 25/50
 277/277 42s 86ms/step -
 accuracy: 0.7603 - loss: 0.5597 - val_accuracy: 0.7545 - val_loss: 0.5589

Epoch 26/50
 277/277 38s 75ms/step -
 accuracy: 0.7603 - loss: 0.5583 - val_accuracy: 0.7545 - val_loss: 0.5587

Epoch 27/50
 277/277 41s 74ms/step -
 accuracy: 0.7603 - loss: 0.5600 - val_accuracy: 0.7545 - val_loss: 0.5588

Epoch 28/50
 277/277 41s 74ms/step -
 accuracy: 0.7603 - loss: 0.5591 - val_accuracy: 0.7545 - val_loss: 0.5586

Epoch 29/50
 277/277 44s 87ms/step -
 accuracy: 0.7603 - loss: 0.5576 - val_accuracy: 0.7545 - val_loss: 0.5585

Epoch 30/50
 277/277 20s 73ms/step -
 accuracy: 0.7603 - loss: 0.5579 - val_accuracy: 0.7545 - val_loss: 0.5585

Epoch 31/50
 277/277 21s 73ms/step -
 accuracy: 0.7603 - loss: 0.5591 - val_accuracy: 0.7545 - val_loss: 0.5585

Epoch 32/50
 277/277 23s 81ms/step -
 accuracy: 0.7603 - loss: 0.5577 - val_accuracy: 0.7545 - val_loss: 0.5583

Epoch 33/50
 277/277 40s 77ms/step -
 accuracy: 0.7603 - loss: 0.5575 - val_accuracy: 0.7545 - val_loss: 0.5585

Epoch 34/50
 277/277 40s 75ms/step -
 accuracy: 0.7603 - loss: 0.5579 - val_accuracy: 0.7545 - val_loss: 0.5584

Epoch 35/50
 277/277 24s 87ms/step -
 accuracy: 0.7603 - loss: 0.5576 - val_accuracy: 0.7545 - val_loss: 0.5582

Epoch 36/50
 277/277 37s 74ms/step -
 accuracy: 0.7603 - loss: 0.5569 - val_accuracy: 0.7545 - val_loss: 0.5586

Epoch 37/50
 277/277 22s 81ms/step -
 accuracy: 0.7603 - loss: 0.5583 - val_accuracy: 0.7545 - val_loss: 0.5582

Epoch 38/50
 277/277 41s 82ms/step -
 accuracy: 0.7603 - loss: 0.5581 - val_accuracy: 0.7545 - val_loss: 0.5581

Epoch 39/50
 277/277 39s 75ms/step -
 accuracy: 0.7603 - loss: 0.5566 - val_accuracy: 0.7545 - val_loss: 0.5582

Epoch 40/50
 277/277 41s 74ms/step -
 accuracy: 0.7603 - loss: 0.5574 - val_accuracy: 0.7545 - val_loss: 0.5582

Epoch 41/50
 277/277 41s 74ms/step -
 accuracy: 0.7603 - loss: 0.5569 - val_accuracy: 0.7545 - val_loss: 0.5580

Epoch 42/50
 277/277 41s 74ms/step -
 accuracy: 0.7603 - loss: 0.5570 - val_accuracy: 0.7545 - val_loss: 0.5581


```

Epoch 43/50
277/277          41s 74ms/step -
accuracy: 0.7603 - loss: 0.5569 - val_accuracy: 0.7545 - val_loss: 0.5580
Epoch 44/50
277/277          43s 81ms/step -
accuracy: 0.7603 - loss: 0.5565 - val_accuracy: 0.7545 - val_loss: 0.5580
Epoch 45/50
277/277          39s 75ms/step -
accuracy: 0.7603 - loss: 0.5549 - val_accuracy: 0.7545 - val_loss: 0.5579
Epoch 46/50
277/277          41s 75ms/step -
accuracy: 0.7603 - loss: 0.5546 - val_accuracy: 0.7545 - val_loss: 0.5579
Epoch 47/50
277/277          23s 82ms/step -
accuracy: 0.7603 - loss: 0.5573 - val_accuracy: 0.7545 - val_loss: 0.5579
Epoch 48/50
277/277          40s 79ms/step -
accuracy: 0.7603 - loss: 0.5572 - val_accuracy: 0.7545 - val_loss: 0.5579
Epoch 49/50
277/277          40s 74ms/step -
accuracy: 0.7603 - loss: 0.5540 - val_accuracy: 0.7545 - val_loss: 0.5578
Epoch 50/50
277/277          41s 74ms/step -
accuracy: 0.7603 - loss: 0.5572 - val_accuracy: 0.7545 - val_loss: 0.5578
139/139          1s 9ms/step
Fold 3 - Precision: 0.7545, Recall: 1.0000, F1 Score: 0.8601
Training for fold 4 on GossipCop...
Epoch 1/50

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(

277/277          25s 81ms/step -
accuracy: 0.6056 - loss: 1.7161 - val_accuracy: 0.7642 - val_loss: 1.0730
Epoch 2/50
277/277          20s 73ms/step -
accuracy: 0.7587 - loss: 0.9947 - val_accuracy: 0.7642 - val_loss: 0.7518
Epoch 3/50
277/277          21s 75ms/step -
accuracy: 0.7587 - loss: 0.7374 - val_accuracy: 0.7642 - val_loss: 0.6294
Epoch 4/50
277/277          40s 73ms/step -
accuracy: 0.7587 - loss: 0.6346 - val_accuracy: 0.7642 - val_loss: 0.5842
Epoch 5/50
277/277          22s 80ms/step -
accuracy: 0.7587 - loss: 0.5963 - val_accuracy: 0.7642 - val_loss: 0.5672
Epoch 6/50
277/277          40s 77ms/step -

```

accuracy: 0.7587 - loss: 0.5834 - val_accuracy: 0.7642 - val_loss: 0.5590
 Epoch 7/50
 277/277 42s 79ms/step -
 accuracy: 0.7587 - loss: 0.5773 - val_accuracy: 0.7642 - val_loss: 0.5568
 Epoch 8/50
 277/277 40s 76ms/step -
 accuracy: 0.7587 - loss: 0.5733 - val_accuracy: 0.7642 - val_loss: 0.5537
 Epoch 9/50
 277/277 23s 83ms/step -
 accuracy: 0.7587 - loss: 0.5747 - val_accuracy: 0.7642 - val_loss: 0.5540
 Epoch 10/50
 277/277 39s 75ms/step -
 accuracy: 0.7587 - loss: 0.5712 - val_accuracy: 0.7642 - val_loss: 0.5526
 Epoch 11/50
 277/277 41s 74ms/step -
 accuracy: 0.7587 - loss: 0.5723 - val_accuracy: 0.7642 - val_loss: 0.5519
 Epoch 12/50
 277/277 41s 74ms/step -
 accuracy: 0.7587 - loss: 0.5684 - val_accuracy: 0.7642 - val_loss: 0.5511
 Epoch 13/50
 277/277 23s 82ms/step -
 accuracy: 0.7587 - loss: 0.5684 - val_accuracy: 0.7642 - val_loss: 0.5511
 Epoch 14/50
 277/277 39s 77ms/step -
 accuracy: 0.7587 - loss: 0.5695 - val_accuracy: 0.7642 - val_loss: 0.5509
 Epoch 15/50
 277/277 40s 74ms/step -
 accuracy: 0.7587 - loss: 0.5650 - val_accuracy: 0.7642 - val_loss: 0.5497
 Epoch 16/50
 277/277 22s 80ms/step -
 accuracy: 0.7587 - loss: 0.5686 - val_accuracy: 0.7642 - val_loss: 0.5505
 Epoch 17/50
 277/277 40s 78ms/step -
 accuracy: 0.7587 - loss: 0.5675 - val_accuracy: 0.7642 - val_loss: 0.5495
 Epoch 18/50
 277/277 42s 83ms/step -
 accuracy: 0.7587 - loss: 0.5686 - val_accuracy: 0.7642 - val_loss: 0.5496
 Epoch 19/50
 277/277 39s 74ms/step -
 accuracy: 0.7587 - loss: 0.5655 - val_accuracy: 0.7642 - val_loss: 0.5491
 Epoch 20/50
 277/277 41s 74ms/step -
 accuracy: 0.7587 - loss: 0.5653 - val_accuracy: 0.7642 - val_loss: 0.5496
 Epoch 21/50
 277/277 41s 73ms/step -
 accuracy: 0.7587 - loss: 0.5634 - val_accuracy: 0.7642 - val_loss: 0.5494
 Epoch 22/50
 277/277 23s 84ms/step -

accuracy: 0.7587 - loss: 0.5661 - val_accuracy: 0.7642 - val_loss: 0.5494
 Epoch 23/50
 277/277 41s 82ms/step -
 accuracy: 0.7587 - loss: 0.5639 - val_accuracy: 0.7642 - val_loss: 0.5487
 Epoch 24/50
 277/277 22s 80ms/step -
 accuracy: 0.7587 - loss: 0.5625 - val_accuracy: 0.7642 - val_loss: 0.5486
 Epoch 25/50
 277/277 42s 82ms/step -
 accuracy: 0.7587 - loss: 0.5641 - val_accuracy: 0.7642 - val_loss: 0.5484
 Epoch 26/50
 277/277 39s 75ms/step -
 accuracy: 0.7587 - loss: 0.5640 - val_accuracy: 0.7642 - val_loss: 0.5482
 Epoch 27/50
 277/277 41s 75ms/step -
 accuracy: 0.7587 - loss: 0.5636 - val_accuracy: 0.7642 - val_loss: 0.5486
 Epoch 28/50
 277/277 43s 83ms/step -
 accuracy: 0.7587 - loss: 0.5639 - val_accuracy: 0.7642 - val_loss: 0.5485
 Epoch 29/50
 277/277 39s 74ms/step -
 accuracy: 0.7587 - loss: 0.5628 - val_accuracy: 0.7642 - val_loss: 0.5489
 Epoch 30/50
 277/277 41s 74ms/step -
 accuracy: 0.7587 - loss: 0.5627 - val_accuracy: 0.7642 - val_loss: 0.5485
 Epoch 31/50
 277/277 41s 73ms/step -
 accuracy: 0.7587 - loss: 0.5616 - val_accuracy: 0.7642 - val_loss: 0.5483
 Epoch 32/50
 277/277 24s 84ms/step -
 accuracy: 0.7587 - loss: 0.5629 - val_accuracy: 0.7642 - val_loss: 0.5481
 Epoch 33/50
 277/277 39s 76ms/step -
 accuracy: 0.7587 - loss: 0.5612 - val_accuracy: 0.7642 - val_loss: 0.5478
 Epoch 34/50
 277/277 41s 75ms/step -
 accuracy: 0.7587 - loss: 0.5620 - val_accuracy: 0.7642 - val_loss: 0.5478
 Epoch 35/50
 277/277 22s 79ms/step -
 accuracy: 0.7587 - loss: 0.5602 - val_accuracy: 0.7642 - val_loss: 0.5479
 Epoch 36/50
 277/277 42s 83ms/step -
 accuracy: 0.7587 - loss: 0.5609 - val_accuracy: 0.7642 - val_loss: 0.5485
 Epoch 37/50
 277/277 39s 75ms/step -
 accuracy: 0.7587 - loss: 0.5599 - val_accuracy: 0.7642 - val_loss: 0.5479
 Epoch 38/50
 277/277 41s 75ms/step -

```

accuracy: 0.7587 - loss: 0.5603 - val_accuracy: 0.7642 - val_loss: 0.5477
Epoch 39/50
277/277          22s 79ms/step -
accuracy: 0.7587 - loss: 0.5603 - val_accuracy: 0.7642 - val_loss: 0.5475
Epoch 40/50
277/277          20s 73ms/step -
accuracy: 0.7587 - loss: 0.5589 - val_accuracy: 0.7642 - val_loss: 0.5472
Epoch 41/50
277/277          22s 79ms/step -
accuracy: 0.7587 - loss: 0.5615 - val_accuracy: 0.7642 - val_loss: 0.5480
Epoch 42/50
277/277          40s 75ms/step -
accuracy: 0.7587 - loss: 0.5583 - val_accuracy: 0.7642 - val_loss: 0.5479
Epoch 43/50
277/277          41s 74ms/step -
accuracy: 0.7587 - loss: 0.5580 - val_accuracy: 0.7642 - val_loss: 0.5484
Epoch 44/50
277/277          24s 87ms/step -
accuracy: 0.7587 - loss: 0.5593 - val_accuracy: 0.7642 - val_loss: 0.5483
Epoch 45/50
277/277          37s 75ms/step -
accuracy: 0.7587 - loss: 0.5598 - val_accuracy: 0.7642 - val_loss: 0.5477
Epoch 46/50
277/277          41s 76ms/step -
accuracy: 0.7587 - loss: 0.5576 - val_accuracy: 0.7642 - val_loss: 0.5478
Epoch 47/50
277/277          40s 74ms/step -
accuracy: 0.7587 - loss: 0.5594 - val_accuracy: 0.7642 - val_loss: 0.5478
Epoch 48/50
277/277          44s 84ms/step -
accuracy: 0.7587 - loss: 0.5591 - val_accuracy: 0.7642 - val_loss: 0.5474
Epoch 49/50
277/277          22s 78ms/step -
accuracy: 0.7587 - loss: 0.5597 - val_accuracy: 0.7642 - val_loss: 0.5475
Epoch 50/50
277/277          41s 79ms/step -
accuracy: 0.7587 - loss: 0.5564 - val_accuracy: 0.7642 - val_loss: 0.5477
139/139          1s 9ms/step
Fold 4 - Precision: 0.7642, Recall: 1.0000, F1 Score: 0.8664
Training for fold 5 on GossipCop...
Epoch 1/50

```

```

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(

```

```

277/277          25s 77ms/step -
accuracy: 0.7521 - loss: 1.6654 - val_accuracy: 0.7574 - val_loss: 1.0783
Epoch 2/50

```

277/277 21s 74ms/step -
accuracy: 0.7600 - loss: 0.9843 - val_accuracy: 0.7574 - val_loss: 0.7640
Epoch 3/50

277/277 41s 74ms/step -
accuracy: 0.7600 - loss: 0.7298 - val_accuracy: 0.7574 - val_loss: 0.6398
Epoch 4/50

277/277 25s 91ms/step -
accuracy: 0.7600 - loss: 0.6332 - val_accuracy: 0.7574 - val_loss: 0.5928
Epoch 5/50

277/277 36s 73ms/step -
accuracy: 0.7600 - loss: 0.5925 - val_accuracy: 0.7574 - val_loss: 0.5743
Epoch 6/50

277/277 23s 83ms/step -
accuracy: 0.7600 - loss: 0.5766 - val_accuracy: 0.7574 - val_loss: 0.5626
Epoch 7/50

277/277 39s 76ms/step -
accuracy: 0.7612 - loss: 0.5553 - val_accuracy: 0.7990 - val_loss: 0.4870
Epoch 8/50

277/277 41s 74ms/step -
accuracy: 0.8111 - loss: 0.4608 - val_accuracy: 0.8240 - val_loss: 0.4558
Epoch 9/50

277/277 43s 81ms/step -
accuracy: 0.8420 - loss: 0.4137 - val_accuracy: 0.8301 - val_loss: 0.4451
Epoch 10/50

277/277 39s 75ms/step -
accuracy: 0.8617 - loss: 0.3873 - val_accuracy: 0.8306 - val_loss: 0.4416
Epoch 11/50

277/277 40s 73ms/step -
accuracy: 0.8726 - loss: 0.3649 - val_accuracy: 0.8308 - val_loss: 0.4407
Epoch 12/50

277/277 23s 83ms/step -
accuracy: 0.8819 - loss: 0.3469 - val_accuracy: 0.8335 - val_loss: 0.4441
Epoch 13/50

277/277 39s 75ms/step -
accuracy: 0.8858 - loss: 0.3388 - val_accuracy: 0.8333 - val_loss: 0.4484
Epoch 14/50

277/277 24s 88ms/step -
accuracy: 0.8935 - loss: 0.3237 - val_accuracy: 0.8326 - val_loss: 0.4516
Epoch 15/50

277/277 20s 73ms/step -
accuracy: 0.8968 - loss: 0.3165 - val_accuracy: 0.8331 - val_loss: 0.4565
Epoch 16/50

277/277 21s 76ms/step -
accuracy: 0.9011 - loss: 0.3065 - val_accuracy: 0.8342 - val_loss: 0.4605
Epoch 17/50

277/277 22s 78ms/step -
accuracy: 0.9056 - loss: 0.2997 - val_accuracy: 0.8306 - val_loss: 0.4676
Epoch 18/50

277/277 42s 80ms/step -
 accuracy: 0.9061 - loss: 0.2904 - val_accuracy: 0.8313 - val_loss: 0.4735
 Epoch 19/50
 277/277 40s 75ms/step -
 accuracy: 0.9107 - loss: 0.2856 - val_accuracy: 0.8292 - val_loss: 0.4821
 Epoch 20/50
 277/277 43s 84ms/step -
 accuracy: 0.9138 - loss: 0.2779 - val_accuracy: 0.8292 - val_loss: 0.4897
 Epoch 21/50
 277/277 38s 74ms/step -
 accuracy: 0.9144 - loss: 0.2747 - val_accuracy: 0.8295 - val_loss: 0.4983
 Epoch 22/50
 277/277 23s 84ms/step -
 accuracy: 0.9191 - loss: 0.2687 - val_accuracy: 0.8279 - val_loss: 0.5039
 Epoch 23/50
 277/277 20s 74ms/step -
 accuracy: 0.9235 - loss: 0.2609 - val_accuracy: 0.8274 - val_loss: 0.5135
 Epoch 24/50
 277/277 21s 75ms/step -
 accuracy: 0.9247 - loss: 0.2556 - val_accuracy: 0.8240 - val_loss: 0.5172
 Epoch 25/50
 277/277 41s 75ms/step -
 accuracy: 0.9271 - loss: 0.2524 - val_accuracy: 0.8254 - val_loss: 0.5246
 Epoch 26/50
 277/277 43s 83ms/step -
 accuracy: 0.9283 - loss: 0.2463 - val_accuracy: 0.8215 - val_loss: 0.5353
 Epoch 27/50
 277/277 39s 75ms/step -
 accuracy: 0.9301 - loss: 0.2439 - val_accuracy: 0.8227 - val_loss: 0.5398
 Epoch 28/50
 277/277 41s 74ms/step -
 accuracy: 0.9316 - loss: 0.2375 - val_accuracy: 0.8234 - val_loss: 0.5515
 Epoch 29/50
 277/277 21s 75ms/step -
 accuracy: 0.9352 - loss: 0.2306 - val_accuracy: 0.8222 - val_loss: 0.5577
 Epoch 30/50
 277/277 41s 75ms/step -
 accuracy: 0.9373 - loss: 0.2250 - val_accuracy: 0.8227 - val_loss: 0.5723
 Epoch 31/50
 277/277 42s 80ms/step -
 accuracy: 0.9394 - loss: 0.2196 - val_accuracy: 0.8225 - val_loss: 0.5828
 Epoch 32/50
 277/277 39s 75ms/step -
 accuracy: 0.9446 - loss: 0.2169 - val_accuracy: 0.8225 - val_loss: 0.5937
 Epoch 33/50
 277/277 41s 75ms/step -
 accuracy: 0.9439 - loss: 0.2116 - val_accuracy: 0.8211 - val_loss: 0.5984
 Epoch 34/50

277/277 41s 75ms/step -
 accuracy: 0.9481 - loss: 0.2057 - val_accuracy: 0.8197 - val_loss: 0.6167
 Epoch 35/50
 277/277 41s 74ms/step -
 accuracy: 0.9488 - loss: 0.2012 - val_accuracy: 0.8193 - val_loss: 0.6277
 Epoch 36/50
 277/277 41s 75ms/step -
 accuracy: 0.9527 - loss: 0.1956 - val_accuracy: 0.8179 - val_loss: 0.6435
 Epoch 37/50
 277/277 41s 74ms/step -
 accuracy: 0.9564 - loss: 0.1889 - val_accuracy: 0.8173 - val_loss: 0.6620
 Epoch 38/50
 277/277 41s 74ms/step -
 accuracy: 0.9563 - loss: 0.1848 - val_accuracy: 0.8175 - val_loss: 0.6725
 Epoch 39/50
 277/277 41s 74ms/step -
 accuracy: 0.9592 - loss: 0.1831 - val_accuracy: 0.8161 - val_loss: 0.6842
 Epoch 40/50
 277/277 43s 82ms/step -
 accuracy: 0.9610 - loss: 0.1767 - val_accuracy: 0.8164 - val_loss: 0.6948
 Epoch 41/50
 277/277 22s 81ms/step -
 accuracy: 0.9631 - loss: 0.1732 - val_accuracy: 0.8161 - val_loss: 0.7112
 Epoch 42/50
 277/277 40s 76ms/step -
 accuracy: 0.9636 - loss: 0.1693 - val_accuracy: 0.8159 - val_loss: 0.7305
 Epoch 43/50
 277/277 22s 79ms/step -
 accuracy: 0.9665 - loss: 0.1635 - val_accuracy: 0.8157 - val_loss: 0.7394
 Epoch 44/50
 277/277 41s 81ms/step -
 accuracy: 0.9692 - loss: 0.1593 - val_accuracy: 0.8157 - val_loss: 0.7501
 Epoch 45/50
 277/277 42s 84ms/step -
 accuracy: 0.9691 - loss: 0.1557 - val_accuracy: 0.8139 - val_loss: 0.7634
 Epoch 46/50
 277/277 22s 80ms/step -
 accuracy: 0.9693 - loss: 0.1520 - val_accuracy: 0.8139 - val_loss: 0.7789
 Epoch 47/50
 277/277 41s 79ms/step -
 accuracy: 0.9724 - loss: 0.1476 - val_accuracy: 0.8166 - val_loss: 0.8010
 Epoch 48/50
 277/277 41s 80ms/step -
 accuracy: 0.9718 - loss: 0.1435 - val_accuracy: 0.8173 - val_loss: 0.8234
 Epoch 49/50
 277/277 39s 74ms/step -
 accuracy: 0.9736 - loss: 0.1396 - val_accuracy: 0.8150 - val_loss: 0.8232
 Epoch 50/50

```

277/277          22s 81ms/step -
accuracy: 0.9732 - loss: 0.1386 - val_accuracy: 0.8116 - val_loss: 0.8188
139/139          1s 10ms/step
Fold 5 - Precision: 0.8675, Recall: 0.8867, F1 Score: 0.8770
Average Precision: 0.8242
Average Recall: 0.9412
Average F1 Score: 0.8759
Training for fold 1 on PolitiFact...
Epoch 1/50

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(

14/14           6s 139ms/step -
accuracy: 0.5793 - loss: 3.4624 - val_accuracy: 0.6161 - val_loss: 3.3594
Epoch 2/50
14/14           2s 116ms/step -
accuracy: 0.5982 - loss: 3.3315 - val_accuracy: 0.6161 - val_loss: 3.2334
Epoch 3/50
14/14           2s 112ms/step -
accuracy: 0.5910 - loss: 3.2054 - val_accuracy: 0.6161 - val_loss: 3.1132
Epoch 4/50
14/14           2s 112ms/step -
accuracy: 0.5936 - loss: 3.0865 - val_accuracy: 0.6161 - val_loss: 2.9989
Epoch 5/50
14/14           2s 117ms/step -
accuracy: 0.5896 - loss: 2.9759 - val_accuracy: 0.6161 - val_loss: 2.8906
Epoch 6/50
14/14           2s 171ms/step -
accuracy: 0.5880 - loss: 2.8654 - val_accuracy: 0.6161 - val_loss: 2.7875
Epoch 7/50
14/14           3s 184ms/step -
accuracy: 0.5869 - loss: 2.7640 - val_accuracy: 0.6161 - val_loss: 2.6897
Epoch 8/50
14/14           4s 120ms/step -
accuracy: 0.5874 - loss: 2.6670 - val_accuracy: 0.6161 - val_loss: 2.5964
Epoch 9/50
14/14           2s 116ms/step -
accuracy: 0.5874 - loss: 2.5766 - val_accuracy: 0.6161 - val_loss: 2.5080
Epoch 10/50
14/14           2s 112ms/step -
accuracy: 0.5904 - loss: 2.4879 - val_accuracy: 0.6161 - val_loss: 2.4241
Epoch 11/50
14/14           3s 175ms/step -
accuracy: 0.5882 - loss: 2.4035 - val_accuracy: 0.6161 - val_loss: 2.3440
Epoch 12/50
14/14           2s 175ms/step -
accuracy: 0.5894 - loss: 2.3246 - val_accuracy: 0.6161 - val_loss: 2.2676

```


Epoch 13/50
14/14 2s 113ms/step -
accuracy: 0.5892 - loss: 2.2535 - val_accuracy: 0.6161 - val_loss: 2.1956
Epoch 14/50
14/14 2s 115ms/step -
accuracy: 0.5936 - loss: 2.1781 - val_accuracy: 0.6161 - val_loss: 2.1268
Epoch 15/50
14/14 3s 117ms/step -
accuracy: 0.5924 - loss: 2.1102 - val_accuracy: 0.6161 - val_loss: 2.0611
Epoch 16/50
14/14 2s 117ms/step -
accuracy: 0.5931 - loss: 2.0443 - val_accuracy: 0.6161 - val_loss: 1.9985
Epoch 17/50
14/14 3s 120ms/step -
accuracy: 0.5903 - loss: 1.9839 - val_accuracy: 0.6161 - val_loss: 1.9391
Epoch 18/50
14/14 3s 170ms/step -
accuracy: 0.5906 - loss: 1.9259 - val_accuracy: 0.6161 - val_loss: 1.8826
Epoch 19/50
14/14 2s 138ms/step -
accuracy: 0.5909 - loss: 1.8688 - val_accuracy: 0.6161 - val_loss: 1.8285
Epoch 20/50
14/14 2s 114ms/step -
accuracy: 0.5922 - loss: 1.8115 - val_accuracy: 0.6161 - val_loss: 1.7771
Epoch 21/50
14/14 2s 117ms/step -
accuracy: 0.5918 - loss: 1.7648 - val_accuracy: 0.6161 - val_loss: 1.7280
Epoch 22/50
14/14 2s 119ms/step -
accuracy: 0.5971 - loss: 1.7133 - val_accuracy: 0.6161 - val_loss: 1.6810
Epoch 23/50
14/14 2s 113ms/step -
accuracy: 0.6190 - loss: 1.6669 - val_accuracy: 0.6161 - val_loss: 1.6368
Epoch 24/50
14/14 2s 129ms/step -
accuracy: 0.6374 - loss: 1.6226 - val_accuracy: 0.6161 - val_loss: 1.5944
Epoch 25/50
14/14 3s 185ms/step -
accuracy: 0.6555 - loss: 1.5774 - val_accuracy: 0.6161 - val_loss: 1.5528
Epoch 26/50
14/14 4s 118ms/step -
accuracy: 0.6578 - loss: 1.5365 - val_accuracy: 0.6161 - val_loss: 1.5130
Epoch 27/50
14/14 3s 116ms/step -
accuracy: 0.6838 - loss: 1.4930 - val_accuracy: 0.6493 - val_loss: 1.4747
Epoch 28/50
14/14 2s 112ms/step -
accuracy: 0.7153 - loss: 1.4526 - val_accuracy: 0.6682 - val_loss: 1.4375

Epoch 29/50
14/14 3s 148ms/step -
accuracy: 0.7424 - loss: 1.4145 - val_accuracy: 0.6872 - val_loss: 1.4004
Epoch 30/50
14/14 3s 175ms/step -
accuracy: 0.7691 - loss: 1.3732 - val_accuracy: 0.7251 - val_loss: 1.3637
Epoch 31/50
14/14 2s 124ms/step -
accuracy: 0.7980 - loss: 1.3360 - val_accuracy: 0.7488 - val_loss: 1.3276
Epoch 32/50
14/14 2s 117ms/step -
accuracy: 0.8040 - loss: 1.2906 - val_accuracy: 0.7536 - val_loss: 1.2915
Epoch 33/50
14/14 3s 115ms/step -
accuracy: 0.8465 - loss: 1.2452 - val_accuracy: 0.7678 - val_loss: 1.2556
Epoch 34/50
14/14 2s 117ms/step -
accuracy: 0.8608 - loss: 1.2008 - val_accuracy: 0.7773 - val_loss: 1.2198
Epoch 35/50
14/14 3s 143ms/step -
accuracy: 0.8880 - loss: 1.1477 - val_accuracy: 0.7962 - val_loss: 1.1844
Epoch 36/50
14/14 3s 181ms/step -
accuracy: 0.8939 - loss: 1.0910 - val_accuracy: 0.7962 - val_loss: 1.1501
Epoch 37/50
14/14 4s 117ms/step -
accuracy: 0.8984 - loss: 1.0541 - val_accuracy: 0.8104 - val_loss: 1.1183
Epoch 38/50
14/14 2s 113ms/step -
accuracy: 0.9046 - loss: 1.0022 - val_accuracy: 0.8152 - val_loss: 1.0908
Epoch 39/50
14/14 2s 117ms/step -
accuracy: 0.9266 - loss: 0.9479 - val_accuracy: 0.8199 - val_loss: 1.0656
Epoch 40/50
14/14 3s 135ms/step -
accuracy: 0.9411 - loss: 0.9188 - val_accuracy: 0.8199 - val_loss: 1.0435
Epoch 41/50
14/14 3s 185ms/step -
accuracy: 0.9499 - loss: 0.8638 - val_accuracy: 0.8341 - val_loss: 1.0215
Epoch 42/50
14/14 4s 112ms/step -
accuracy: 0.9476 - loss: 0.8538 - val_accuracy: 0.8246 - val_loss: 1.0043
Epoch 43/50
14/14 2s 113ms/step -
accuracy: 0.9549 - loss: 0.8111 - val_accuracy: 0.8246 - val_loss: 0.9870
Epoch 44/50
14/14 3s 121ms/step -
accuracy: 0.9600 - loss: 0.7987 - val_accuracy: 0.8294 - val_loss: 0.9715

```

Epoch 45/50
14/14          2s 117ms/step -
accuracy: 0.9647 - loss: 0.7633 - val_accuracy: 0.8341 - val_loss: 0.9556
Epoch 46/50
14/14          2s 151ms/step -
accuracy: 0.9636 - loss: 0.7371 - val_accuracy: 0.8389 - val_loss: 0.9350
Epoch 47/50
14/14          3s 176ms/step -
accuracy: 0.9664 - loss: 0.7146 - val_accuracy: 0.8483 - val_loss: 0.9256
Epoch 48/50
14/14          2s 116ms/step -
accuracy: 0.9691 - loss: 0.6973 - val_accuracy: 0.8436 - val_loss: 0.9109
Epoch 49/50
14/14          2s 117ms/step -
accuracy: 0.9635 - loss: 0.6743 - val_accuracy: 0.8483 - val_loss: 0.8954
Epoch 50/50
14/14          2s 117ms/step -
accuracy: 0.9720 - loss: 0.6500 - val_accuracy: 0.8578 - val_loss: 0.8830
7/7            0s 30ms/step
Fold 1 - Precision: 0.8472, Recall: 0.9385, F1 Score: 0.8905
Training for fold 2 on PolitiFact...
Epoch 1/50

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(

14/14          5s 157ms/step -
accuracy: 0.5861 - loss: 3.4623 - val_accuracy: 0.6066 - val_loss: 3.3619
Epoch 2/50
14/14          3s 183ms/step -
accuracy: 0.5967 - loss: 3.3321 - val_accuracy: 0.6066 - val_loss: 3.2362
Epoch 3/50
14/14          2s 141ms/step -
accuracy: 0.5877 - loss: 3.2075 - val_accuracy: 0.6066 - val_loss: 3.1163
Epoch 4/50
14/14          2s 111ms/step -
accuracy: 0.5822 - loss: 3.0897 - val_accuracy: 0.6066 - val_loss: 3.0020
Epoch 5/50
14/14          2s 112ms/step -
accuracy: 0.5840 - loss: 2.9763 - val_accuracy: 0.6066 - val_loss: 2.8932
Epoch 6/50
14/14          2s 111ms/step -
accuracy: 0.5892 - loss: 2.8688 - val_accuracy: 0.6066 - val_loss: 2.7898
Epoch 7/50
14/14          3s 113ms/step -
accuracy: 0.5844 - loss: 2.7663 - val_accuracy: 0.6066 - val_loss: 2.6915
Epoch 8/50
14/14          2s 114ms/step -

```

accuracy: 0.5841 - loss: 2.6685 - val_accuracy: 0.6066 - val_loss: 2.5978
 Epoch 9/50
 14/14 2s 167ms/step -
 accuracy: 0.5839 - loss: 2.5763 - val_accuracy: 0.6066 - val_loss: 2.5089
 Epoch 10/50
 14/14 3s 182ms/step -
 accuracy: 0.5833 - loss: 2.4913 - val_accuracy: 0.6066 - val_loss: 2.4247
 Epoch 11/50
 14/14 4s 115ms/step -
 accuracy: 0.5829 - loss: 2.4094 - val_accuracy: 0.6066 - val_loss: 2.3448
 Epoch 12/50
 14/14 2s 113ms/step -
 accuracy: 0.5833 - loss: 2.3281 - val_accuracy: 0.6066 - val_loss: 2.2685
 Epoch 13/50
 14/14 2s 113ms/step -
 accuracy: 0.5830 - loss: 2.2545 - val_accuracy: 0.6066 - val_loss: 2.1961
 Epoch 14/50
 14/14 2s 117ms/step -
 accuracy: 0.5863 - loss: 2.1807 - val_accuracy: 0.6066 - val_loss: 2.1272
 Epoch 15/50
 14/14 3s 167ms/step -
 accuracy: 0.5835 - loss: 2.1132 - val_accuracy: 0.6066 - val_loss: 2.0618
 Epoch 16/50
 14/14 3s 186ms/step -
 accuracy: 0.5855 - loss: 2.0467 - val_accuracy: 0.6066 - val_loss: 1.9994
 Epoch 17/50
 14/14 4s 115ms/step -
 accuracy: 0.5840 - loss: 1.9884 - val_accuracy: 0.6066 - val_loss: 1.9401
 Epoch 18/50
 14/14 2s 111ms/step -
 accuracy: 0.5848 - loss: 1.9275 - val_accuracy: 0.6066 - val_loss: 1.8839
 Epoch 19/50
 14/14 2s 117ms/step -
 accuracy: 0.5830 - loss: 1.8733 - val_accuracy: 0.6066 - val_loss: 1.8302
 Epoch 20/50
 14/14 2s 111ms/step -
 accuracy: 0.5852 - loss: 1.8186 - val_accuracy: 0.6066 - val_loss: 1.7792
 Epoch 21/50
 14/14 3s 169ms/step -
 accuracy: 0.5861 - loss: 1.7723 - val_accuracy: 0.6066 - val_loss: 1.7309
 Epoch 22/50
 14/14 3s 199ms/step -
 accuracy: 0.5878 - loss: 1.7228 - val_accuracy: 0.6066 - val_loss: 1.6848
 Epoch 23/50
 14/14 4s 115ms/step -
 accuracy: 0.5898 - loss: 1.6755 - val_accuracy: 0.6066 - val_loss: 1.6407
 Epoch 24/50
 14/14 3s 113ms/step -

accuracy: 0.5894 - loss: 1.6321 - val_accuracy: 0.6066 - val_loss: 1.5986
 Epoch 25/50
 14/14 3s 116ms/step -
 accuracy: 0.5875 - loss: 1.5901 - val_accuracy: 0.6066 - val_loss: 1.5586
 Epoch 26/50
 14/14 2s 111ms/step -
 accuracy: 0.5879 - loss: 1.5541 - val_accuracy: 0.6066 - val_loss: 1.5206
 Epoch 27/50
 14/14 2s 175ms/step -
 accuracy: 0.5860 - loss: 1.5160 - val_accuracy: 0.6066 - val_loss: 1.4844
 Epoch 28/50
 14/14 3s 184ms/step -
 accuracy: 0.5855 - loss: 1.4758 - val_accuracy: 0.6066 - val_loss: 1.4497
 Epoch 29/50
 14/14 4s 114ms/step -
 accuracy: 0.5890 - loss: 1.4457 - val_accuracy: 0.6066 - val_loss: 1.4167
 Epoch 30/50
 14/14 3s 121ms/step -
 accuracy: 0.5940 - loss: 1.4129 - val_accuracy: 0.6066 - val_loss: 1.3853
 Epoch 31/50
 14/14 2s 115ms/step -
 accuracy: 0.6031 - loss: 1.3820 - val_accuracy: 0.6066 - val_loss: 1.3549
 Epoch 32/50
 14/14 3s 171ms/step -
 accuracy: 0.5984 - loss: 1.3513 - val_accuracy: 0.6066 - val_loss: 1.3258
 Epoch 33/50
 14/14 3s 173ms/step -
 accuracy: 0.6105 - loss: 1.3154 - val_accuracy: 0.6066 - val_loss: 1.2972
 Epoch 34/50
 14/14 2s 118ms/step -
 accuracy: 0.6243 - loss: 1.2926 - val_accuracy: 0.6066 - val_loss: 1.2698
 Epoch 35/50
 14/14 2s 115ms/step -
 accuracy: 0.6533 - loss: 1.2595 - val_accuracy: 0.6114 - val_loss: 1.2427
 Epoch 36/50
 14/14 2s 113ms/step -
 accuracy: 0.6815 - loss: 1.2342 - val_accuracy: 0.6445 - val_loss: 1.2159
 Epoch 37/50
 14/14 2s 118ms/step -
 accuracy: 0.7274 - loss: 1.1986 - val_accuracy: 0.7441 - val_loss: 1.1882
 Epoch 38/50
 14/14 2s 114ms/step -
 accuracy: 0.7520 - loss: 1.1724 - val_accuracy: 0.7867 - val_loss: 1.1592
 Epoch 39/50
 14/14 2s 110ms/step -
 accuracy: 0.8034 - loss: 1.1306 - val_accuracy: 0.8294 - val_loss: 1.1282
 Epoch 40/50
 14/14 3s 172ms/step -

```

accuracy: 0.8286 - loss: 1.0948 - val_accuracy: 0.8436 - val_loss: 1.0932
Epoch 41/50
14/14          2s 170ms/step -
accuracy: 0.8605 - loss: 1.0548 - val_accuracy: 0.8578 - val_loss: 1.0562
Epoch 42/50
14/14          2s 113ms/step -
accuracy: 0.8864 - loss: 0.9946 - val_accuracy: 0.8673 - val_loss: 1.0169
Epoch 43/50
14/14          3s 115ms/step -
accuracy: 0.9043 - loss: 0.9448 - val_accuracy: 0.8768 - val_loss: 0.9771
Epoch 44/50
14/14          3s 120ms/step -
accuracy: 0.9278 - loss: 0.8802 - val_accuracy: 0.8626 - val_loss: 0.9395
Epoch 45/50
14/14          2s 114ms/step -
accuracy: 0.9391 - loss: 0.8303 - val_accuracy: 0.8720 - val_loss: 0.9049
Epoch 46/50
14/14          2s 115ms/step -
accuracy: 0.9398 - loss: 0.7910 - val_accuracy: 0.8720 - val_loss: 0.8745
Epoch 47/50
14/14          2s 172ms/step -
accuracy: 0.9517 - loss: 0.7573 - val_accuracy: 0.8768 - val_loss: 0.8514
Epoch 48/50
14/14          3s 185ms/step -
accuracy: 0.9589 - loss: 0.7222 - val_accuracy: 0.8626 - val_loss: 0.8303
Epoch 49/50
14/14          2s 117ms/step -
accuracy: 0.9468 - loss: 0.6982 - val_accuracy: 0.8720 - val_loss: 0.8147
Epoch 50/50
14/14          3s 117ms/step -
accuracy: 0.9640 - loss: 0.6687 - val_accuracy: 0.8720 - val_loss: 0.8020

```

```

WARNING:tensorflow:5 out of the last 147 calls to <function
TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed at
0x792c7929de10> triggered tf.function retracing. Tracing is expensive and the
excessive number of tracings could be due to (1) creating @tf.function
repeatedly in a loop, (2) passing tensors with different shapes, (3) passing
Python objects instead of tensors. For (1), please define your @tf.function
outside of the loop. For (2), @tf.function has reduce_retracing=True option that
can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling\_retracing and
https://www.tensorflow.org/api\_docs/python/tf/function for more details.

```

```

7/7          0s 33ms/step
Fold 2 - Precision: 0.8976, Recall: 0.8906, F1 Score: 0.8941
Training for fold 3 on PolitiFact...
Epoch 1/50

```

```

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:

```

UserWarning: Argument `input_length` is deprecated. Just remove it.
warnings.warn(

```
14/14          5s 149ms/step -  
accuracy: 0.5788 - loss: 3.4714 - val_accuracy: 0.5640 - val_loss: 3.3727  
Epoch 2/50  
14/14          3s 184ms/step -  
accuracy: 0.6267 - loss: 3.3393 - val_accuracy: 0.5640 - val_loss: 3.2472  
Epoch 3/50  
14/14          4s 114ms/step -  
accuracy: 0.6010 - loss: 3.2146 - val_accuracy: 0.5640 - val_loss: 3.1274  
Epoch 4/50  
14/14          3s 116ms/step -  
accuracy: 0.6072 - loss: 3.0934 - val_accuracy: 0.5640 - val_loss: 3.0132  
Epoch 5/50  
14/14          2s 115ms/step -  
accuracy: 0.6119 - loss: 2.9814 - val_accuracy: 0.5640 - val_loss: 2.9048  
Epoch 6/50  
14/14          2s 112ms/step -  
accuracy: 0.6073 - loss: 2.8725 - val_accuracy: 0.5640 - val_loss: 2.8017  
Epoch 7/50  
14/14          2s 169ms/step -  
accuracy: 0.6055 - loss: 2.7701 - val_accuracy: 0.5640 - val_loss: 2.7038  
Epoch 8/50  
14/14          3s 185ms/step -  
accuracy: 0.6060 - loss: 2.6749 - val_accuracy: 0.5640 - val_loss: 2.6106  
Epoch 9/50  
14/14          4s 116ms/step -  
accuracy: 0.6001 - loss: 2.5817 - val_accuracy: 0.5640 - val_loss: 2.5222  
Epoch 10/50  
14/14          3s 114ms/step -  
accuracy: 0.6092 - loss: 2.4949 - val_accuracy: 0.5640 - val_loss: 2.4381  
Epoch 11/50  
14/14          2s 118ms/step -  
accuracy: 0.6044 - loss: 2.4083 - val_accuracy: 0.5640 - val_loss: 2.3584  
Epoch 12/50  
14/14          3s 134ms/step -  
accuracy: 0.6072 - loss: 2.3301 - val_accuracy: 0.5640 - val_loss: 2.2823  
Epoch 13/50  
14/14          3s 188ms/step -  
accuracy: 0.6065 - loss: 2.2516 - val_accuracy: 0.5640 - val_loss: 2.2102  
Epoch 14/50  
14/14          4s 116ms/step -  
accuracy: 0.6012 - loss: 2.1827 - val_accuracy: 0.5640 - val_loss: 2.1415  
Epoch 15/50  
14/14          2s 114ms/step -  
accuracy: 0.6061 - loss: 2.1123 - val_accuracy: 0.5640 - val_loss: 2.0763  
Epoch 16/50
```

14/14 3s 116ms/step -
 accuracy: 0.6085 - loss: 2.0466 - val_accuracy: 0.5640 - val_loss: 2.0146
 Epoch 17/50
 14/14 2s 116ms/step -
 accuracy: 0.6056 - loss: 1.9851 - val_accuracy: 0.5640 - val_loss: 1.9560
 Epoch 18/50
 14/14 2s 170ms/step -
 accuracy: 0.6042 - loss: 1.9287 - val_accuracy: 0.5640 - val_loss: 1.8996
 Epoch 19/50
 14/14 3s 183ms/step -
 accuracy: 0.6064 - loss: 1.8712 - val_accuracy: 0.5640 - val_loss: 1.8459
 Epoch 20/50
 14/14 2s 121ms/step -
 accuracy: 0.6050 - loss: 1.8213 - val_accuracy: 0.5640 - val_loss: 1.7949
 Epoch 21/50
 14/14 2s 113ms/step -
 accuracy: 0.6095 - loss: 1.7673 - val_accuracy: 0.5640 - val_loss: 1.7463
 Epoch 22/50
 14/14 3s 116ms/step -
 accuracy: 0.6042 - loss: 1.7152 - val_accuracy: 0.5640 - val_loss: 1.6999
 Epoch 23/50
 14/14 3s 117ms/step -
 accuracy: 0.6145 - loss: 1.6759 - val_accuracy: 0.5640 - val_loss: 1.6555
 Epoch 24/50
 14/14 2s 137ms/step -
 accuracy: 0.6351 - loss: 1.6228 - val_accuracy: 0.5640 - val_loss: 1.6133
 Epoch 25/50
 14/14 3s 182ms/step -
 accuracy: 0.6358 - loss: 1.5857 - val_accuracy: 0.5640 - val_loss: 1.5731
 Epoch 26/50
 14/14 2s 157ms/step -
 accuracy: 0.6319 - loss: 1.5464 - val_accuracy: 0.5640 - val_loss: 1.5340
 Epoch 27/50
 14/14 2s 113ms/step -
 accuracy: 0.6569 - loss: 1.5016 - val_accuracy: 0.5640 - val_loss: 1.4964
 Epoch 28/50
 14/14 2s 113ms/step -
 accuracy: 0.6878 - loss: 1.4626 - val_accuracy: 0.5735 - val_loss: 1.4601
 Epoch 29/50
 14/14 2s 116ms/step -
 accuracy: 0.7247 - loss: 1.4194 - val_accuracy: 0.6351 - val_loss: 1.4246
 Epoch 30/50
 14/14 2s 118ms/step -
 accuracy: 0.7363 - loss: 1.3826 - val_accuracy: 0.7014 - val_loss: 1.3900
 Epoch 31/50
 14/14 2s 112ms/step -
 accuracy: 0.7433 - loss: 1.3402 - val_accuracy: 0.7393 - val_loss: 1.3554
 Epoch 32/50

14/14 3s 171ms/step -
 accuracy: 0.8000 - loss: 1.2934 - val_accuracy: 0.7536 - val_loss: 1.3211
 Epoch 33/50
 14/14 3s 174ms/step -
 accuracy: 0.7977 - loss: 1.2500 - val_accuracy: 0.7583 - val_loss: 1.2870
 Epoch 34/50
 14/14 2s 112ms/step -
 accuracy: 0.8516 - loss: 1.1916 - val_accuracy: 0.7773 - val_loss: 1.2528
 Epoch 35/50
 14/14 2s 113ms/step -
 accuracy: 0.8617 - loss: 1.1455 - val_accuracy: 0.7630 - val_loss: 1.2196
 Epoch 36/50
 14/14 3s 116ms/step -
 accuracy: 0.8969 - loss: 1.0924 - val_accuracy: 0.7867 - val_loss: 1.1881
 Epoch 37/50
 14/14 3s 114ms/step -
 accuracy: 0.9081 - loss: 1.0433 - val_accuracy: 0.7962 - val_loss: 1.1596
 Epoch 38/50
 14/14 3s 158ms/step -
 accuracy: 0.9283 - loss: 0.9854 - val_accuracy: 0.8009 - val_loss: 1.1346
 Epoch 39/50
 14/14 3s 185ms/step -
 accuracy: 0.9295 - loss: 0.9427 - val_accuracy: 0.8009 - val_loss: 1.1121
 Epoch 40/50
 14/14 4s 112ms/step -
 accuracy: 0.9483 - loss: 0.9036 - val_accuracy: 0.8057 - val_loss: 1.0926
 Epoch 41/50
 14/14 2s 112ms/step -
 accuracy: 0.9438 - loss: 0.8696 - val_accuracy: 0.8057 - val_loss: 1.0753
 Epoch 42/50
 14/14 3s 118ms/step -
 accuracy: 0.9518 - loss: 0.8452 - val_accuracy: 0.8152 - val_loss: 1.0584
 Epoch 43/50
 14/14 2s 113ms/step -
 accuracy: 0.9624 - loss: 0.7980 - val_accuracy: 0.8009 - val_loss: 1.0449
 Epoch 44/50
 14/14 2s 182ms/step -
 accuracy: 0.9632 - loss: 0.7851 - val_accuracy: 0.8104 - val_loss: 1.0296
 Epoch 45/50
 14/14 2s 175ms/step -
 accuracy: 0.9616 - loss: 0.7684 - val_accuracy: 0.8152 - val_loss: 1.0131
 Epoch 46/50
 14/14 2s 122ms/step -
 accuracy: 0.9587 - loss: 0.7375 - val_accuracy: 0.8057 - val_loss: 1.0000
 Epoch 47/50
 14/14 3s 142ms/step -
 accuracy: 0.9697 - loss: 0.7110 - val_accuracy: 0.8104 - val_loss: 0.9861
 Epoch 48/50

```

14/14          3s 186ms/step -
accuracy: 0.9663 - loss: 0.7014 - val_accuracy: 0.8152 - val_loss: 0.9707
Epoch 49/50
14/14          2s 120ms/step -
accuracy: 0.9676 - loss: 0.6806 - val_accuracy: 0.8199 - val_loss: 0.9531
Epoch 50/50
14/14          2s 135ms/step -
accuracy: 0.9674 - loss: 0.6791 - val_accuracy: 0.8199 - val_loss: 0.9374

WARNING:tensorflow:5 out of the last 15 calls to <function
TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed at
0x792c7930e050> triggered tf.function retracing. Tracing is expensive and the
excessive number of tracings could be due to (1) creating @tf.function
repeatedly in a loop, (2) passing tensors with different shapes, (3) passing
Python objects instead of tensors. For (1), please define your @tf.function
outside of the loop. For (2), @tf.function has reduce_retracing=True option that
can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling\_retracing and
https://www.tensorflow.org/api\_docs/python/tf/function for more details.

7/7           0s 44ms/step
Fold 3 - Precision: 0.8293, Recall: 0.8571, F1 Score: 0.8430
Training for fold 4 on PolitiFact...
Epoch 1/50

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(

14/14          5s 132ms/step -
accuracy: 0.5130 - loss: 3.4579 - val_accuracy: 0.5687 - val_loss: 3.3594
Epoch 2/50
14/14          2s 115ms/step -
accuracy: 0.5385 - loss: 3.3281 - val_accuracy: 0.5782 - val_loss: 3.2340
Epoch 3/50
14/14          2s 114ms/step -
accuracy: 0.5679 - loss: 3.2035 - val_accuracy: 0.5782 - val_loss: 3.1142
Epoch 4/50
14/14          3s 114ms/step -
accuracy: 0.6048 - loss: 3.0829 - val_accuracy: 0.5782 - val_loss: 2.9997
Epoch 5/50
14/14          2s 116ms/step -
accuracy: 0.6016 - loss: 2.9689 - val_accuracy: 0.5782 - val_loss: 2.8907
Epoch 6/50
14/14          3s 184ms/step -
accuracy: 0.6038 - loss: 2.8629 - val_accuracy: 0.5782 - val_loss: 2.7872
Epoch 7/50
14/14          3s 192ms/step -
accuracy: 0.5966 - loss: 2.7609 - val_accuracy: 0.5782 - val_loss: 2.6890
Epoch 8/50

```

14/14 2s 114ms/step -
 accuracy: 0.5976 - loss: 2.6615 - val_accuracy: 0.5782 - val_loss: 2.5956
 Epoch 9/50
 14/14 2s 112ms/step -
 accuracy: 0.6021 - loss: 2.5714 - val_accuracy: 0.5782 - val_loss: 2.5070
 Epoch 10/50
 14/14 2s 112ms/step -
 accuracy: 0.6041 - loss: 2.4825 - val_accuracy: 0.5782 - val_loss: 2.4224
 Epoch 11/50
 14/14 3s 112ms/step -
 accuracy: 0.6014 - loss: 2.3971 - val_accuracy: 0.5782 - val_loss: 2.3422
 Epoch 12/50
 14/14 3s 114ms/step -
 accuracy: 0.6000 - loss: 2.3201 - val_accuracy: 0.5782 - val_loss: 2.2658
 Epoch 13/50
 14/14 3s 169ms/step -
 accuracy: 0.6024 - loss: 2.2443 - val_accuracy: 0.5782 - val_loss: 2.1930
 Epoch 14/50
 14/14 2s 144ms/step -
 accuracy: 0.6064 - loss: 2.1730 - val_accuracy: 0.5782 - val_loss: 2.1239
 Epoch 15/50
 14/14 2s 111ms/step -
 accuracy: 0.5982 - loss: 2.1045 - val_accuracy: 0.5782 - val_loss: 2.0584
 Epoch 16/50
 14/14 3s 117ms/step -
 accuracy: 0.6068 - loss: 2.0380 - val_accuracy: 0.5782 - val_loss: 1.9960
 Epoch 17/50
 14/14 3s 117ms/step -
 accuracy: 0.5997 - loss: 1.9784 - val_accuracy: 0.5782 - val_loss: 1.9367
 Epoch 18/50
 14/14 2s 113ms/step -
 accuracy: 0.5999 - loss: 1.9165 - val_accuracy: 0.5782 - val_loss: 1.8802
 Epoch 19/50
 14/14 2s 171ms/step -
 accuracy: 0.5995 - loss: 1.8595 - val_accuracy: 0.5782 - val_loss: 1.8265
 Epoch 20/50
 14/14 3s 175ms/step -
 accuracy: 0.5998 - loss: 1.8066 - val_accuracy: 0.5782 - val_loss: 1.7754
 Epoch 21/50
 14/14 2s 115ms/step -
 accuracy: 0.5998 - loss: 1.7550 - val_accuracy: 0.5782 - val_loss: 1.7269
 Epoch 22/50
 14/14 2s 117ms/step -
 accuracy: 0.5989 - loss: 1.7091 - val_accuracy: 0.5782 - val_loss: 1.6807
 Epoch 23/50
 14/14 2s 117ms/step -
 accuracy: 0.6022 - loss: 1.6640 - val_accuracy: 0.5782 - val_loss: 1.6366
 Epoch 24/50

14/14 2s 114ms/step -
 accuracy: 0.6012 - loss: 1.6242 - val_accuracy: 0.5782 - val_loss: 1.5947
 Epoch 25/50
 14/14 2s 111ms/step -
 accuracy: 0.5989 - loss: 1.5791 - val_accuracy: 0.5782 - val_loss: 1.5548
 Epoch 26/50
 14/14 2s 111ms/step -
 accuracy: 0.6072 - loss: 1.5400 - val_accuracy: 0.5782 - val_loss: 1.5168
 Epoch 27/50
 14/14 2s 158ms/step -
 accuracy: 0.6031 - loss: 1.5003 - val_accuracy: 0.5782 - val_loss: 1.4805
 Epoch 28/50
 14/14 3s 176ms/step -
 accuracy: 0.6025 - loss: 1.4647 - val_accuracy: 0.5782 - val_loss: 1.4458
 Epoch 29/50
 14/14 2s 114ms/step -
 accuracy: 0.6027 - loss: 1.4327 - val_accuracy: 0.5782 - val_loss: 1.4126
 Epoch 30/50
 14/14 2s 117ms/step -
 accuracy: 0.6066 - loss: 1.3964 - val_accuracy: 0.5782 - val_loss: 1.3805
 Epoch 31/50
 14/14 2s 117ms/step -
 accuracy: 0.6128 - loss: 1.3615 - val_accuracy: 0.5782 - val_loss: 1.3500
 Epoch 32/50
 14/14 3s 114ms/step -
 accuracy: 0.6197 - loss: 1.3359 - val_accuracy: 0.5782 - val_loss: 1.3200
 Epoch 33/50
 14/14 2s 115ms/step -
 accuracy: 0.6204 - loss: 1.3113 - val_accuracy: 0.5782 - val_loss: 1.2907
 Epoch 34/50
 14/14 3s 171ms/step -
 accuracy: 0.6544 - loss: 1.2764 - val_accuracy: 0.5877 - val_loss: 1.2620
 Epoch 35/50
 14/14 3s 182ms/step -
 accuracy: 0.7094 - loss: 1.2427 - val_accuracy: 0.6019 - val_loss: 1.2331
 Epoch 36/50
 14/14 4s 111ms/step -
 accuracy: 0.7292 - loss: 1.2125 - val_accuracy: 0.6540 - val_loss: 1.2036
 Epoch 37/50
 14/14 3s 112ms/step -
 accuracy: 0.7756 - loss: 1.1734 - val_accuracy: 0.7062 - val_loss: 1.1718
 Epoch 38/50
 14/14 3s 116ms/step -
 accuracy: 0.8125 - loss: 1.1429 - val_accuracy: 0.8294 - val_loss: 1.1375
 Epoch 39/50
 14/14 3s 181ms/step -
 accuracy: 0.8559 - loss: 1.0959 - val_accuracy: 0.8483 - val_loss: 1.0997
 Epoch 40/50

```

14/14          3s 184ms/step -
accuracy: 0.8913 - loss: 1.0554 - val_accuracy: 0.8720 - val_loss: 1.0584
Epoch 41/50
14/14          4s 113ms/step -
accuracy: 0.9179 - loss: 0.9913 - val_accuracy: 0.8815 - val_loss: 1.0142
Epoch 42/50
14/14          2s 113ms/step -
accuracy: 0.9273 - loss: 0.9354 - val_accuracy: 0.8910 - val_loss: 0.9691
Epoch 43/50
14/14          3s 118ms/step -
accuracy: 0.9284 - loss: 0.8721 - val_accuracy: 0.8863 - val_loss: 0.9289
Epoch 44/50
14/14          2s 110ms/step -
accuracy: 0.9412 - loss: 0.8235 - val_accuracy: 0.8910 - val_loss: 0.8955
Epoch 45/50
14/14          2s 169ms/step -
accuracy: 0.9519 - loss: 0.7766 - val_accuracy: 0.8957 - val_loss: 0.8697
Epoch 46/50
14/14          3s 187ms/step -
accuracy: 0.9556 - loss: 0.7446 - val_accuracy: 0.8910 - val_loss: 0.8468
Epoch 47/50
14/14          2s 114ms/step -
accuracy: 0.9519 - loss: 0.7133 - val_accuracy: 0.8910 - val_loss: 0.8295
Epoch 48/50
14/14          2s 112ms/step -
accuracy: 0.9625 - loss: 0.7010 - val_accuracy: 0.8910 - val_loss: 0.8131
Epoch 49/50
14/14          2s 113ms/step -
accuracy: 0.9626 - loss: 0.6633 - val_accuracy: 0.8910 - val_loss: 0.7984
Epoch 50/50
14/14          2s 113ms/step -
accuracy: 0.9698 - loss: 0.6470 - val_accuracy: 0.8863 - val_loss: 0.7850
7/7           0s 29ms/step
Fold 4 - Precision: 0.8769, Recall: 0.9344, F1 Score: 0.9048
Training for fold 5 on PolitiFact...
Epoch 1/50

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(

14/14          6s 201ms/step -
accuracy: 0.5680 - loss: 3.4520 - val_accuracy: 0.5857 - val_loss: 3.3528
Epoch 2/50
14/14          2s 115ms/step -
accuracy: 0.5995 - loss: 3.3196 - val_accuracy: 0.5857 - val_loss: 3.2278
Epoch 3/50
14/14          3s 117ms/step -
accuracy: 0.6112 - loss: 3.1962 - val_accuracy: 0.5857 - val_loss: 3.1090

```

Epoch 4/50
14/14 2s 112ms/step -
accuracy: 0.6152 - loss: 3.0774 - val_accuracy: 0.5857 - val_loss: 2.9966
Epoch 5/50
14/14 3s 118ms/step -
accuracy: 0.6066 - loss: 2.9667 - val_accuracy: 0.5857 - val_loss: 2.8898
Epoch 6/50
14/14 2s 113ms/step -
accuracy: 0.6172 - loss: 2.8593 - val_accuracy: 0.5857 - val_loss: 2.7881
Epoch 7/50
14/14 2s 175ms/step -
accuracy: 0.6086 - loss: 2.7582 - val_accuracy: 0.5857 - val_loss: 2.6914
Epoch 8/50
14/14 3s 188ms/step -
accuracy: 0.6123 - loss: 2.6621 - val_accuracy: 0.5857 - val_loss: 2.5995
Epoch 9/50
14/14 2s 112ms/step -
accuracy: 0.6211 - loss: 2.5705 - val_accuracy: 0.5857 - val_loss: 2.5121
Epoch 10/50
14/14 2s 112ms/step -
accuracy: 0.6219 - loss: 2.4845 - val_accuracy: 0.5857 - val_loss: 2.4290
Epoch 11/50
14/14 2s 117ms/step -
accuracy: 0.6151 - loss: 2.4029 - val_accuracy: 0.5857 - val_loss: 2.3498
Epoch 12/50
14/14 2s 113ms/step -
accuracy: 0.6224 - loss: 2.3226 - val_accuracy: 0.5857 - val_loss: 2.2744
Epoch 13/50
14/14 2s 117ms/step -
accuracy: 0.6147 - loss: 2.2477 - val_accuracy: 0.5857 - val_loss: 2.2028
Epoch 14/50
14/14 2s 166ms/step -
accuracy: 0.6256 - loss: 2.1761 - val_accuracy: 0.5857 - val_loss: 2.1344
Epoch 15/50
14/14 3s 184ms/step -
accuracy: 0.6243 - loss: 2.1076 - val_accuracy: 0.5857 - val_loss: 2.0694
Epoch 16/50
14/14 4s 112ms/step -
accuracy: 0.6103 - loss: 2.0447 - val_accuracy: 0.5857 - val_loss: 2.0076
Epoch 17/50
14/14 2s 116ms/step -
accuracy: 0.6370 - loss: 1.9819 - val_accuracy: 0.5857 - val_loss: 1.9485
Epoch 18/50
14/14 2s 114ms/step -
accuracy: 0.6400 - loss: 1.9225 - val_accuracy: 0.5857 - val_loss: 1.8925
Epoch 19/50
14/14 3s 113ms/step -
accuracy: 0.6413 - loss: 1.8665 - val_accuracy: 0.5857 - val_loss: 1.8390

Epoch 20/50
14/14 3s 173ms/step -
accuracy: 0.6402 - loss: 1.8106 - val_accuracy: 0.5857 - val_loss: 1.7879

Epoch 21/50
14/14 2s 163ms/step -
accuracy: 0.6702 - loss: 1.7611 - val_accuracy: 0.5857 - val_loss: 1.7391

Epoch 22/50
14/14 2s 112ms/step -
accuracy: 0.6766 - loss: 1.7099 - val_accuracy: 0.5905 - val_loss: 1.6925

Epoch 23/50
14/14 2s 112ms/step -
accuracy: 0.6696 - loss: 1.6643 - val_accuracy: 0.5905 - val_loss: 1.6478

Epoch 24/50
14/14 3s 117ms/step -
accuracy: 0.7093 - loss: 1.6142 - val_accuracy: 0.5905 - val_loss: 1.6050

Epoch 25/50
14/14 3s 150ms/step -
accuracy: 0.7123 - loss: 1.5725 - val_accuracy: 0.6286 - val_loss: 1.5640

Epoch 26/50
14/14 3s 193ms/step -
accuracy: 0.7299 - loss: 1.5256 - val_accuracy: 0.6667 - val_loss: 1.5241

Epoch 27/50
14/14 4s 110ms/step -
accuracy: 0.7604 - loss: 1.4823 - val_accuracy: 0.7048 - val_loss: 1.4853

Epoch 28/50
14/14 3s 112ms/step -
accuracy: 0.7941 - loss: 1.4365 - val_accuracy: 0.7238 - val_loss: 1.4474

Epoch 29/50
14/14 2s 114ms/step -
accuracy: 0.8418 - loss: 1.3912 - val_accuracy: 0.7476 - val_loss: 1.4097

Epoch 30/50
14/14 2s 113ms/step -
accuracy: 0.8379 - loss: 1.3507 - val_accuracy: 0.7429 - val_loss: 1.3723

Epoch 31/50
14/14 2s 112ms/step -
accuracy: 0.8492 - loss: 1.3024 - val_accuracy: 0.7571 - val_loss: 1.3344

Epoch 32/50
14/14 2s 131ms/step -
accuracy: 0.8806 - loss: 1.2474 - val_accuracy: 0.7667 - val_loss: 1.2963

Epoch 33/50
14/14 3s 185ms/step -
accuracy: 0.8943 - loss: 1.1923 - val_accuracy: 0.7714 - val_loss: 1.2584

Epoch 34/50
14/14 2s 119ms/step -
accuracy: 0.9179 - loss: 1.1270 - val_accuracy: 0.7810 - val_loss: 1.2221

Epoch 35/50
14/14 2s 115ms/step -
accuracy: 0.9255 - loss: 1.0804 - val_accuracy: 0.7857 - val_loss: 1.1890

Epoch 36/50
14/14 2s 117ms/step -
accuracy: 0.9325 - loss: 1.0235 - val_accuracy: 0.7905 - val_loss: 1.1583
Epoch 37/50
14/14 2s 119ms/step -
accuracy: 0.9487 - loss: 0.9690 - val_accuracy: 0.8000 - val_loss: 1.1303
Epoch 38/50
14/14 2s 113ms/step -
accuracy: 0.9401 - loss: 0.9277 - val_accuracy: 0.8048 - val_loss: 1.1058
Epoch 39/50
14/14 2s 114ms/step -
accuracy: 0.9494 - loss: 0.8895 - val_accuracy: 0.8095 - val_loss: 1.0840
Epoch 40/50
14/14 3s 175ms/step -
accuracy: 0.9522 - loss: 0.8527 - val_accuracy: 0.8048 - val_loss: 1.0640
Epoch 41/50
14/14 2s 162ms/step -
accuracy: 0.9557 - loss: 0.8343 - val_accuracy: 0.8190 - val_loss: 1.0457
Epoch 42/50
14/14 2s 114ms/step -
accuracy: 0.9627 - loss: 0.7917 - val_accuracy: 0.8429 - val_loss: 1.0323
Epoch 43/50
14/14 2s 113ms/step -
accuracy: 0.9655 - loss: 0.7819 - val_accuracy: 0.8429 - val_loss: 1.0170
Epoch 44/50
14/14 2s 115ms/step -
accuracy: 0.9699 - loss: 0.7566 - val_accuracy: 0.8571 - val_loss: 0.9992
Epoch 45/50
14/14 3s 114ms/step -
accuracy: 0.9701 - loss: 0.7284 - val_accuracy: 0.8524 - val_loss: 0.9837
Epoch 46/50
14/14 3s 124ms/step -
accuracy: 0.9622 - loss: 0.7089 - val_accuracy: 0.8524 - val_loss: 0.9705
Epoch 47/50
14/14 2s 172ms/step -
accuracy: 0.9696 - loss: 0.6897 - val_accuracy: 0.8524 - val_loss: 0.9560
Epoch 48/50
14/14 3s 177ms/step -
accuracy: 0.9685 - loss: 0.6736 - val_accuracy: 0.8571 - val_loss: 0.9459
Epoch 49/50
14/14 2s 113ms/step -
accuracy: 0.9681 - loss: 0.6585 - val_accuracy: 0.8571 - val_loss: 0.9383
Epoch 50/50
14/14 2s 118ms/step -
accuracy: 0.9745 - loss: 0.6397 - val_accuracy: 0.8571 - val_loss: 0.9239
7/7 0s 29ms/step
Fold 5 - Precision: 0.8908, Recall: 0.8618, F1 Score: 0.8760
Average Precision: 0.8684

Average Recall: 0.8965
Average F1 Score: 0.8817

The metrics across different folds show stable performance for both datasets:

GossipCop

Fold 1 - Precision: 0.8644, Recall: 0.9141, F1 Score: 0.8885
Fold 2 - Precision: 0.8703, Recall: 0.9051, F1 Score: 0.8873
Fold 3 - Precision: 0.7545, Recall: 1.0000, F1 Score: 0.8601
Fold 4 - Precision: 0.7642, Recall: 1.0000, F1 Score: 0.8664
Fold 5 - Precision: 0.8675, Recall: 0.8867, F1 Score: 0.8770

Average Precision: 0.8242
Average Recall: 0.9412
Average F1 Score: 0.8759

PolitiFact

Fold 1 - Precision: 0.8472, Recall: 0.9385, F1 Score: 0.8905
Fold 2 - Precision: 0.8976, Recall: 0.8906, F1 Score: 0.8941
Fold 3 - Precision: 0.8293, Recall: 0.8571, F1 Score: 0.8430
Fold 4 - Precision: 0.8769, Recall: 0.9344, F1 Score: 0.9048
Fold 5 - Precision: 0.8908, Recall: 0.8618, F1 Score: 0.8760

Average Precision: 0.8648
Average Recall: 0.8965
Average F1 Score: 0.8817

1.10 Hyperparameter Tuning

I have selected Keras tuner for my CNN hyperparameter tuning. Keras Tuner demonstrates superior accuracy in CNN applications (see Table II in Halim et al.[18]).

1.10.1 Hyperparameter Tuning Functions

```
[17]: # This class defines a hypermodel for hyperparameter tuning with Keras Tuner  
# It accepts hyperparameters to tune and inherits from kt.HyperModel and allows  
↪ integration with the Keras Tuner library  
class CNNHyperModel(kt.HyperModel):  
    def __init__(self, input_dim, max_len):  
        # Initialize the hypermodel with the maximum vocabulary size and the  
↪ maximum length of input sequences  
        self.input_dim = input_dim  
        self.max_len = max_len  
  
    def build(self, hp):  
        # Build the model architecture with hyperparameters that will be tuned.  
        # This method is called by the tuner to create new models with  
↪ different hyperparameter values
```

```

    model = Sequential([
        # Embedding layer to transform indices into dense vectors of fixed
        ↪size
        Embedding(input_dim=self.input_dim, output_dim=128,
        ↪input_length=self.max_len),
        # Conv1D layer with hyperparameters for number of filters and
        ↪kernel size that will be tuned
        Conv1D(
            filters=hp.Choice('filters', [32, 64, 128]),
            kernel_size=hp.Choice('kernel_size', [3, 5]),
            activation='relu',
            kernel_regularizer=l2(0.01) # L2 regularization to prevent
            ↪overfitting.
        ),
        MaxPooling1D(pool_size=2), # MaxPooling to reduce the spatial
        ↪dimensions
        GlobalMaxPooling1D(), # Global max pooling to reduce the output of
        ↪the convolutional layer to a single vector
        Dense(
            units=hp.Int('dense_units', min_value=32, max_value=128,
            ↪step=32),
            activation='relu'
        ),
        Dropout(rate=hp.Float('dropout_rate', min_value=0.0, max_value=0.5,
        ↪step=0.1)), # Dropout layer to prevent overfitting
        Dense(1, activation='sigmoid') # Output layer with sigmoid
        ↪activation for binary classification
    ])
    model.compile(
        optimizer=Adam(hp.Float('learning_rate', min_value=1e-4,
        ↪max_value=1e-2, sampling='LOG')),
        loss='binary_crossentropy',
        metrics=[tf.keras.metrics.AUC(curve='PR')] # Use Precision-Recall
        ↪AUC as a metric for imbalanced classification
    )
    return model

# This function configures and executes the hyperparameter tuning using Keras
↪Tuner
# It sets up the tuner, defines the hypermodel, and manages the training process
# Function includes callbacks for early stopping and learning rate reduction
def tune_model(X_train, y_train, X_test, y_test, input_dim, max_len, n_trials,
↪directory, project_name):
    # Setup the hypermodel instance with provided input dimension and maximum
    ↪length
    hypermodel = CNNHyperModel(input_dim=input_dim, max_len=max_len)

```

```

# Configure the tuner with RandomSearch algorithm, focus on maximizing the
↪AUC
tuner = kt.RandomSearch(
    hypermodel,
    objective=kt.Objective("val_auc", direction="max"),
    max_trials=n_trials,
    executions_per_trial=1,
    directory=directory,
    project_name=project_name
)

# Validation data used by tuner
val_data = (X_test, y_test)
# Use existing F1 score callback for additional performance metric tracking
f1_callback = F1ScoreCallback(validation_data=val_data)

# Start hyperparameter search process, use early stopping and learning rate
↪reduction to optimize training
tuner.search(
    x=X_train,
    y=y_train,
    epochs=50,
    validation_data=val_data,
    callbacks=[EarlyStopping(monitor='val_loss', patience=5,
↪restore_best_weights=True),
                ReduceLROnPlateau(monitor='val_loss', factor=0.1,
↪patience=2, min_lr=0.00001),
                f1_callback]
)

# Get the best model after the search
best_model = tuner.get_best_models(num_models=1)[0]
return tuner, best_model # Return the tuner and the best model

```

1.10.2 Hyperparameter Tuning Results

```

[18]: # Run hyperparameter tuning for each dataset
tuner_gossipcop, best_model_gossipcop = tune_model(
    X_train_gossipcop, y_train_gossipcop, X_test_gossipcop, y_test_gossipcop,
    5000, 100, 10, 'hyper_tuning_gossipcop', 'GossipCopTuning'
)

tuner_politifact, best_model_politifact = tune_model(
    X_train_politifact, y_train_politifact, X_test_politifact,
↪y_test_politifact,

```

```
5000, 100, 10, 'hyper_tuning_politifact', 'PolitiFactTuning')
)
```

Trial 10 Complete [00h 00m 16s]

val_auc: 0.9387391209602356

Best val_auc So Far: 0.9391252994537354

Total elapsed time: 00h 06m 14s

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:

UserWarning: Argument `input_length` is deprecated. Just remove it.

warnings.warn(

/usr/local/lib/python3.10/dist-packages/keras/src/saving/saving_lib.py:576:

UserWarning: Skipping variable loading for optimizer 'adam', because it has 2 variables whereas the saved optimizer has 16 variables.

saveable.load_own_variables(weights_store.get(inner_path))

[19]: *# This function displays the best hyperparameters found by the tuner for a specific dataset*

```
def display_best_hyperparameters_and_summary(tuner, dataset_name):
```

```
    # Get the best hyperparameters
```

```
    best_hps = tuner.get_best_hyperparameters()[0]
```

```
    print(f"Best hyperparameters for {dataset_name}:")
```

```
    print(f"  Best filter size: {best_hps.get('filters')}")
```

```
    print(f"  Best kernel size: {best_hps.get('kernel_size')}")
```

```
    print(f"  Best dense units: {best_hps.get('dense_units')}")
```

```
    print(f"  Best learning rate: {best_hps.get('learning_rate')}\n")
```

```
    # Get the best model and print the model summary
```

```
    best_model = tuner.get_best_models(num_models=1)[0]
```

```
    print(f"Model summary for {dataset_name}:")
```

```
    best_model.summary()
```

```
# GossipCop tuning results
```

```
display_best_hyperparameters_and_summary(tuner_gossipcop, "GossipCop")
```

```
# Politifact tuning results
```

```
display_best_hyperparameters_and_summary(tuner_politifact, "PolitiFact")
```

Best hyperparameters for GossipCop:

Best filter size: 32

Best kernel size: 5

Best dense units: 32

Best learning rate: 0.0001655876301640272

Model summary for GossipCop:

Model: "sequential"

Layer (type) ↳ Param #	Output Shape	
embedding (Embedding) ↳ 640,000	(None, 100, 128)	↳
conv1d (Conv1D) ↳ 20,512	(None, 96, 32)	↳
max_pooling1d (MaxPooling1D) ↳ 0	(None, 48, 32)	↳
global_max_pooling1d ↳ 0 (GlobalMaxPooling1D) ↳	(None, 32)	↳
dense (Dense) ↳ 1,056	(None, 32)	↳
dropout (Dropout) ↳ 0	(None, 32)	↳
dense_1 (Dense) ↳ 33	(None, 1)	↳

Total params: 661,601 (2.52 MB)

Trainable params: 661,601 (2.52 MB)

Non-trainable params: 0 (0.00 B)

Best hyperparameters for PolitiFact:

Best filter size: 32

Best kernel size: 5

Best dense units: 64

Best learning rate: 0.0011563133814945622

Model summary for PolitiFact:

Model: "sequential"

Layer (type) ↳ Param #	Output Shape	
embedding (Embedding) ↳ 640,000	(None, 100, 128)	↳
conv1d (Conv1D) ↳ 20,512	(None, 96, 32)	↳
max_pooling1d (MaxPooling1D) ↳ 0	(None, 48, 32)	↳
global_max_pooling1d ↳ 0 (GlobalMaxPooling1D) ↳	(None, 32)	↳
dense (Dense) ↳ 2,112	(None, 64)	↳
dropout (Dropout) ↳ 0	(None, 64)	↳
dense_1 (Dense) ↳ 65	(None, 1)	↳

Total params: 662,689 (2.53 MB)

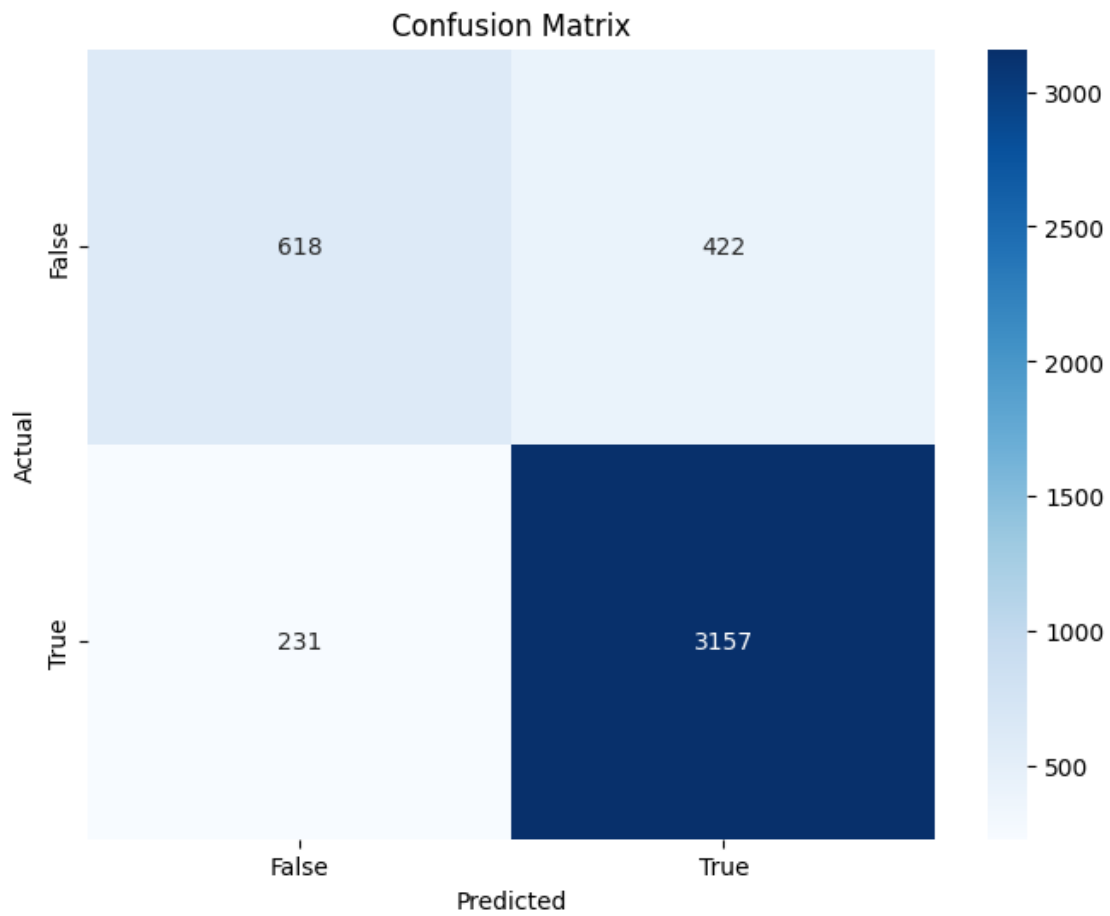
Trainable params: 662,689 (2.53 MB)

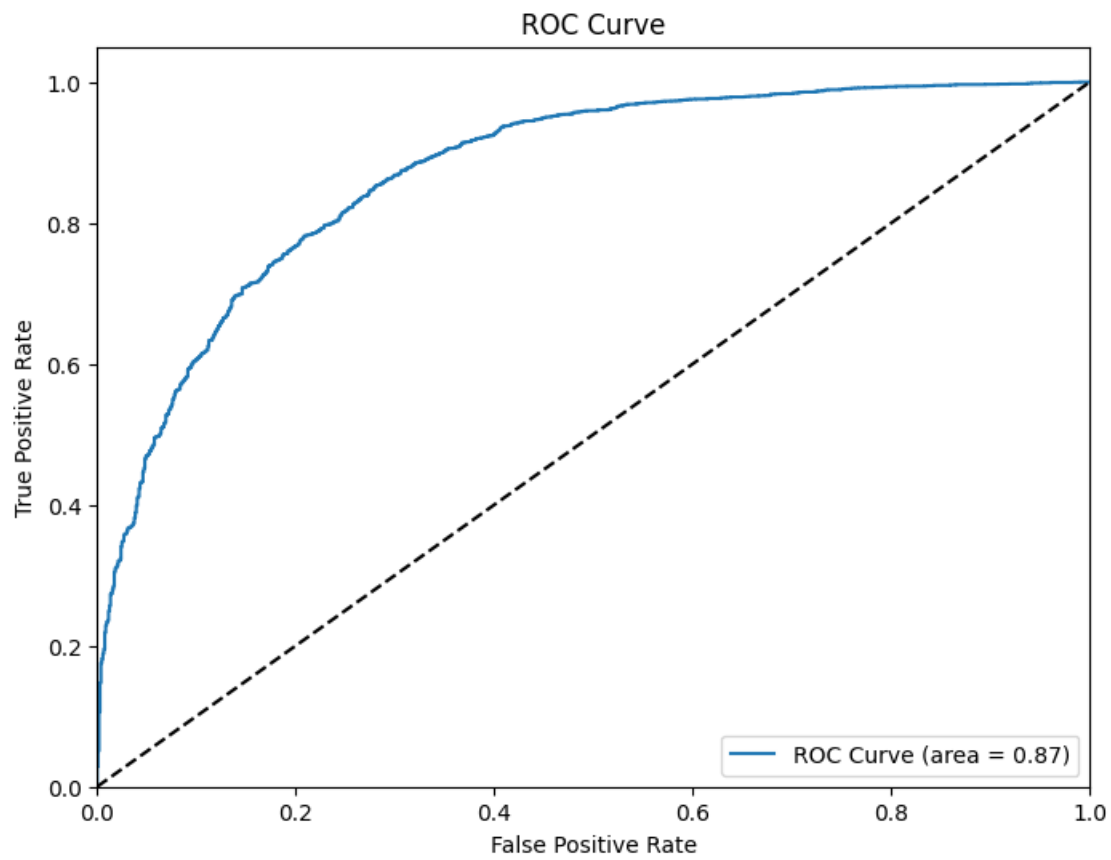
Non-trainable params: 0 (0.00 B)

```
[20]: # Evaluate the tuned model for GossipCop dataset
_, y_pred_gossipcop_tuned, _ = train_and_evaluate(best_model_gossipcop, None,
↳ X_test_gossipcop, None, y_test_gossipcop, None, None, None,
↳ train_model=False)

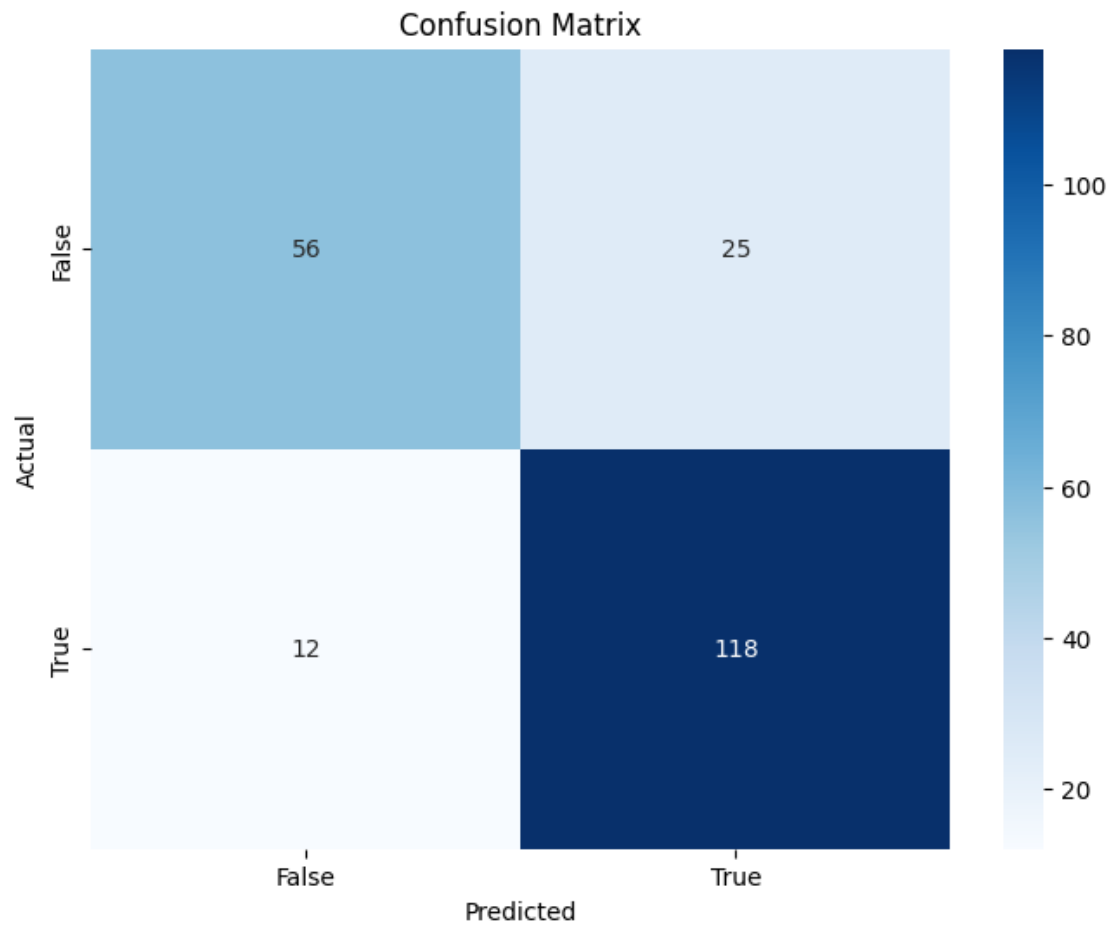
# Evaluate the tuned model for PolitiFact dataset
_, y_pred_politifact_tuned, _ = train_and_evaluate(best_model_politifact, None,
↳ X_test_politifact, None, y_test_politifact, None, None, None,
↳ train_model=False)
```

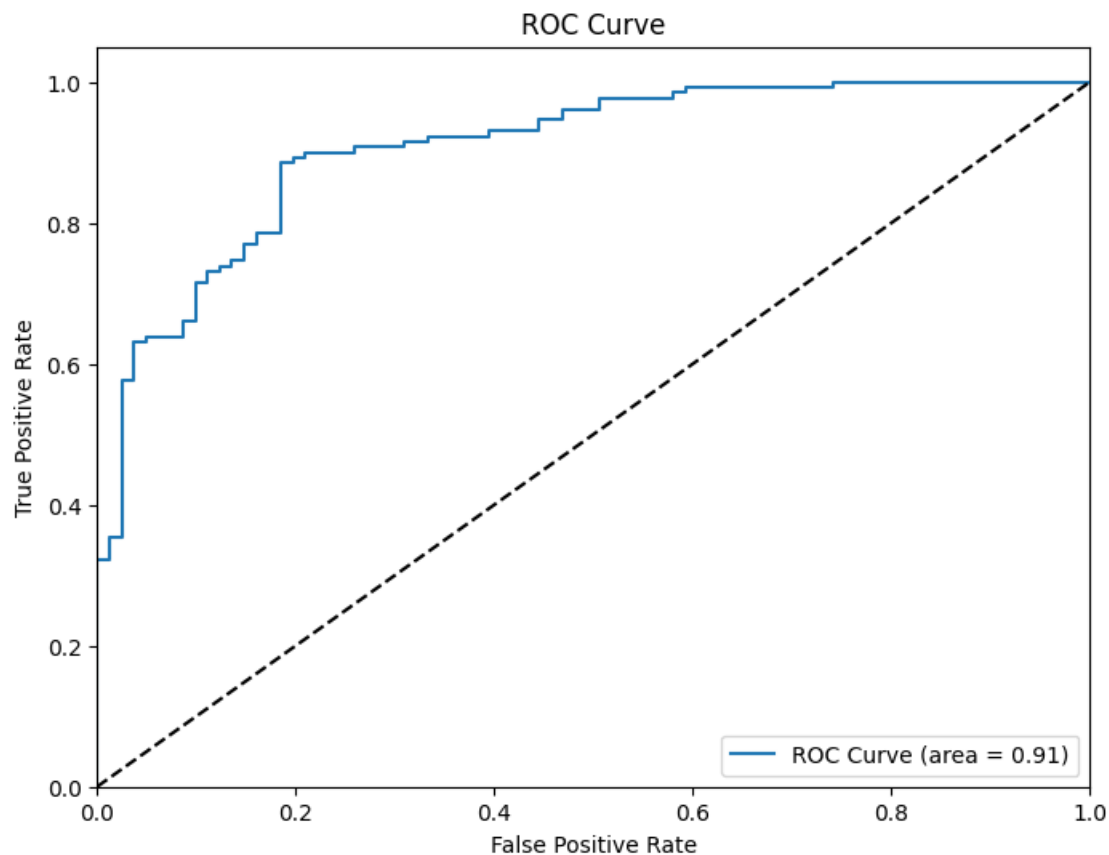
139/139 1s 9ms/step
Accuracy: 0.8525
Precision: 0.8821
Recall: 0.9318
F1-score: 0.9063
ROC AUC: 0.8722
Confusion Matrix:
[[618 422]
 [231 3157]]





7/7 0s 20ms/step
Accuracy: 0.8246
Precision: 0.8252
Recall: 0.9077
F1-score: 0.8645
ROC AUC: 0.9066
Confusion Matrix:
[[56 25]
 [12 118]]





GossipCop Dataset

Initial Model Results:

Accuracy: 0.8318
Precision: 0.8948
Recall: 0.8840
F1-score: 0.8894
ROC AUC: 0.8648

Tuned Model Results:

Accuracy: 0.8525
Precision: 0.8821
Recall: 0.9318
F1-score: 0.9063
ROC AUC: 0.8722

Accuracy and ROC AUC show improvements in the tuned model, with better performance. Recall is higher, which means that it is better at identifying actual positive (fake news) instances. Precision has decreased, so the tuned model has a higher rate of false positives. F1-score is higher.

PolitiFact Dataset

Initial Model Results:

Accuracy: 0.8057
Precision: 0.8504
Recall: 0.8308
F1-score: 0.8405
ROC AUC: 0.8760

Tuned Model Results:

Accuracy: 0.8246
Precision: 0.8252
Recall: 0.9077
F1-score: 0.8645
ROC AUC: 0.9066

Accuracy has increased in the tuned model, this means it is more effective as the initial model. Precision is lower, but recall has increased, this means the tuned model misses more actual positive cases. F1-score is higher and corresponds to the increase in recall. ROC AUC has improved despite the lower accuracy.

For the GossipCop dataset, the tuned model has improved in most metrics, so this is my preferred model.

For the PolitiFact dataset, the results are mixed. Because of this, I choose the original model and not the tuned model.

1.11 Save the Best Models to Google Drive

```
[21]: # Save the best tuned model for GossipCop
best_model_gossipcop = tuner_gossipcop.get_best_models(num_models=1)[0]
model_filename_gossipcop = '/content/drive/My Drive/best_cnn_model_gossipcop.
↳keras'
best_model_gossipcop.save(model_filename_gossipcop)
print("Best CNN model for GossipCop saved to Google Drive")
```

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:

UserWarning: Argument `input_length` is deprecated. Just remove it.

warnings.warn(

/usr/local/lib/python3.10/dist-packages/keras/src/saving/saving_lib.py:576:

UserWarning: Skipping variable loading for optimizer 'adam', because it has 2 variables whereas the saved optimizer has 16 variables.

saveable.load_own_variables(weights_store.get(inner_path))

Best CNN model for GossipCop saved to Google Drive

```
[22]: # Save the GossipCop tokenizer and label encoder
joblib.dump(tokenizer_gossipcop, '/content/drive/My Drive/tokenizer_gossipcop.
↳pk1')
joblib.dump(le_gossipcop, '/content/drive/My Drive/label_encoder_gossipcop.pk1')
print("GossipCop tokenizer and label encoder saved to Google Drive")
```

GossipCop tokenizer and label encoder saved to Google Drive

```
[23]: # Save the best untuned model for PolitiFact
model_filename_politifact = '/content/drive/My Drive/best_cnn_model_politifact.
↳keras'
cnn_model_politifact.save(model_filename_politifact)
print("Best CNN model for PolitiFact saved to Google Drive")
```

Best CNN model for PolitiFact saved to Google Drive

```
[24]: # Save the PolitiFact tokenizer and label encoder
joblib.dump(tokenizer_politifact, '/content/drive/My Drive/tokenizer_politifact.
↳pkl')
joblib.dump(le_politifact, '/content/drive/My Drive/label_encoder_politifact.
↳pkl')
print("PolitiFact tokenizer and label encoder saved to Google Drive")
```

PolitiFact tokenizer and label encoder saved to Google Drive

1.12 Conclusion

My models significantly outperform the baseline results across all metrics for both datasets (GossipCop and PolitiFact). Calculated class weights in both datasets handle class imbalance. Hyperparameter tuning improved GossipCop model performance, which was not the case for PolitiFact model. The best performing model was GossipCop with tuned hyperparameters.

1.13 References

- [7] SHU, K., MAHUDESWARAN, D., WANG, S., LEE, D., and LIU, H. 2018. FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media. <https://doi.org/10.48550/arXiv.1809.01286>
- [11] BRITZ, D. 2021. Convolutional Neural Network for Text Classification in TensorFlow. <https://github.com/dennybritz/cnn-text-classification-tf>
- [15] BROWNE, M.W. 2000. Cross-Validation Methods. Journal of Mathematical Psychology, 44, 108–132. <https://doi.org/10.1006/jmps.1999.1279>
- [16] POWERS, D.M.W. 2011. Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. International Journal of Machine Learning Technology, 2(1), pp. 37-63. <https://doi.org/10.48550/arXiv.2010.16061>
- [18] HALIM, A., CHOW, C., BUDIHARTO, M., ACHMAD, S., and SUTOYO, R. 2023. The Impact of Hyperparameter Tuning in Convolutional Neural Network on Image Classification Model: A Case Study of Plant Disease Detection. In Proceedings of the IEEE Conference on Innovative Research and Development (ICORIS), pp. 1-6. <https://doi.org/10.1109/ICORIS60118.2023.10352209>

```
[ ]: # Install LaTeX packages necessary for converting notebooks to PDF
!apt-get update
```

```

!apt-get install -y texlive-xetex texlive-fonts-recommended_
↳texlive-plain-generic texlive-latex-extra pandoc

# Convert the notebook to PDF
!jupyter nbconvert --to pdf "/content/drive/My Drive/Colab Notebooks/
↳FakeNewsNetCNN.ipynb"

```

```

Get:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
[3,626 B]
Get:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
InRelease [1,581 B]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:5 https://r2u.stat.illinois.edu/ubuntu jammy InRelease
Get:6 https://r2u.stat.illinois.edu/ubuntu jammy Release [5,713 B]
Get:7 https://r2u.stat.illinois.edu/ubuntu jammy Release.gpg [793 B]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:9 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
Packages [976 kB]
Get:10 https://r2u.stat.illinois.edu/ubuntu jammy/main all Packages [8,287 kB]
Hit:11 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2,499
kB]
Get:13 https://r2u.stat.illinois.edu/ubuntu jammy/main amd64 Packages [2,569 kB]
Get:14 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
[18.1 kB]
Hit:15 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy
InRelease
Hit:16 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Get:17 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy/main amd64
Packages [27.8 kB]
Fetched 14.6 MB in 3s (4,836 kB/s)
Reading package lists... Done
W: Skipping acquire of configured file 'main/source/Sources' as repository
'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' does not seem to provide
it (sources.list entry misspelt?)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcbmark-gfm-extensions0.29.0.gfm.3
libcbmark-gfm0.29.0.gfm.3
  libcommons-logging-java libcommons-parent-java libfontbox-java libfontenc1
libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1

```

```

libruby3.0 libsynctex2
  libteckit0 libtexlua53 libtexluajit2 libwoff1 libzzip-0-13 lmodern pandoc-data
poppler-data
  preview-latex-style rake ruby ruby-net-telnet ruby-rubygems ruby-webrick ruby-
xmlrpc ruby3.0
  rubygems-integration tlutils teckit tex-common tex-gyre texlive-base texlive-
binaries
  texlive-latex-base texlive-latex-recommended texlive-pictures tipa xfonts-
encodings xfonts-utils
Suggested packages:
  fonts-noto fonts-freefont-otf | fonts-freefont-ttf libavalon-framework-java
  libcommons-logging-java-doc libexcalibur-logkit-java liblog4j1.2-java texlive-
luatex
  pandoc-citeproc context wkhtmltopdf librsvg2-bin groff ghc nodejs php python
libjs-mathjax
  libjs-katex citation-style-language-styles poppler-utils ghostscript fonts-
japanese-mincho
  | fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic fonts-
arphic-ukai
  fonts-arphic-uming fonts-nanum ri ruby-dev bundler debhelper gv | postscript-
viewer perl-tk xpdf
  | pdf-viewer xzdec texlive-fonts-recommended-doc texlive-latex-base-doc
python3-pygments
  icc-profiles libfile-which-perl libspreadsheet-parseexcel-perl texlive-latex-
extra-doc
  texlive-latex-recommended-doc texlive-pstricks dot2tex prerex texlive-
pictures-doc vprerex
  default-jre-headless tipa-doc
The following NEW packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcmark-gfm-extensions0.29.0.gfm.3
libcmark-gfm0.29.0.gfm.3
  libcommons-logging-java libcommons-parent-java libfontbox-java libfontenc1
libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
libruby3.0 libsynctex2
  libteckit0 libtexlua53 libtexluajit2 libwoff1 libzzip-0-13 lmodern pandoc
pandoc-data
  poppler-data preview-latex-style rake ruby ruby-net-telnet ruby-rubygems ruby-
webrick ruby-xmlrpc
  ruby3.0 rubygems-integration tlutils teckit tex-common tex-gyre texlive-base
texlive-binaries
  texlive-fonts-recommended texlive-latex-base texlive-latex-extra texlive-
latex-recommended
  texlive-pictures texlive-plain-generic texlive-xetex tipa xfonts-encodings
xfonts-utils
0 upgraded, 58 newly installed, 0 to remove and 49 not upgraded.

```

Need to get 202 MB of archives.

After this operation, 728 MB of additional disk space will be used.

Get:1 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 fonts-droid-fallback all 1:6.0.1r16-1.1build1 [1,805 kB]

Get:2 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 fonts-lato all 2.0-2.1 [2,696 kB]

Get:3 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 poppler-data all 0.4.11-1 [2,171 kB]

Get:4 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 tex-common all 6.17 [33.7 kB]

Get:5 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 fonts-urw-base35 all 20200910-1 [6,367 kB]

Get:6 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libgs9-common all 9.55.0~dfsg1-0ubuntu5.9 [752 kB]

Get:7 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libidn12 amd64 1.38-4ubuntu1 [60.0 kB]

Get:8 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 libijs-0.35 amd64 0.35-15build2 [16.5 kB]

Get:9 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 libjbig2dec0 amd64 0.19-3build2 [64.7 kB]

Get:10 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libgs9 amd64 9.55.0~dfsg1-0ubuntu5.9 [5,033 kB]

Get:11 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libkpathsea6 amd64 2021.20210626.59705-1ubuntu0.2 [60.4 kB]

Get:12 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 libwoff1 amd64 1.0.2-1build4 [45.2 kB]

Get:13 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 dvisvgm amd64 2.13.1-1 [1,221 kB]

Get:14 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 fonts-lmodern all 2.004.5-6.1 [4,532 kB]

Get:15 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 fonts-noto-mono all 20201225-1build1 [397 kB]

Get:16 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 fonts-texgyre all 20180621-3.1 [10.2 MB]

Get:17 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libapache-pom-java all 18-1 [4,720 B]

Get:18 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libcmark-gfm0.29.0.gfm.3 amd64 0.29.0.gfm.3-3 [115 kB]

Get:19 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libcmark-gfm-extensions0.29.0.gfm.3 amd64 0.29.0.gfm.3-3 [25.1 kB]

Get:20 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libcommons-parent-java all 43-1 [10.8 kB]

Get:21 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libcommons-logging-java all 1.2-2 [60.3 kB]

Get:22 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 libfontenc1 amd64 1:1.1.4-1build3 [14.7 kB]

Get:23 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libptexenc1 amd64 2021.20210626.59705-1ubuntu0.2 [39.1 kB]

Get:24 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 rubygems-integration all 1.18 [5,336 B]
Get:25 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 ruby3.0 amd64 3.0.2-7ubuntu2.7 [50.1 kB]
Get:26 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 ruby-rubygems all 3.3.5-2 [228 kB]
Get:27 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 ruby amd64 1:3.0~exp1 [5,100 B]
Get:28 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 rake all 13.0.6-2 [61.7 kB]
Get:29 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 ruby-net-telnet all 0.1.1-2 [12.6 kB]
Get:30 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 ruby-webrick all 1.7.0-3 [51.8 kB]
Get:31 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 ruby-xmlrpc all 0.3.2-1ubuntu0.1 [24.9 kB]
Get:32 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libruby3.0 amd64 3.0.2-7ubuntu2.7 [5,113 kB]
Get:33 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libsyntax2 amd64 2021.20210626.59705-1ubuntu0.2 [55.6 kB]
Get:34 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libteckit0 amd64 2.5.11+ds1-1 [421 kB]
Get:35 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libtexlua53 amd64 2021.20210626.59705-1ubuntu0.2 [120 kB]
Get:36 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libtexluajit2 amd64 2021.20210626.59705-1ubuntu0.2 [267 kB]
Get:37 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libzip-0-13 amd64 0.13.72+dfsg.1-1.1 [27.0 kB]
Get:38 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 xfonts-encodings all 1:1.0.5-0ubuntu2 [578 kB]
Get:39 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 xfonts-utils amd64 1:7.7+6build2 [94.6 kB]
Get:40 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 lmodern all 2.004.5-6.1 [9,471 kB]
Get:41 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 pandoc-data all 2.9.2.1-3ubuntu2 [81.8 kB]
Get:42 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 pandoc amd64 2.9.2.1-3ubuntu2 [20.3 MB]
Get:43 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 preview-latex-style all 12.2-1ubuntu1 [185 kB]
Get:44 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 t1utils amd64 1.41-4build2 [61.3 kB]
Get:45 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 teckit amd64 2.5.11+ds1-1 [699 kB]
Get:46 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 tex-gyre all 20180621-3.1 [6,209 kB]
Get:47 <http://archive.ubuntu.com/ubuntu> jammy-updates/universe amd64 texlive-binaries amd64 2021.20210626.59705-1ubuntu0.2 [9,860 kB]


```

Get:48 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-base all
2021.20220204-1 [21.0 MB]
Get:49 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-fonts-
recommended all 2021.20220204-1 [4,972 kB]
Get:50 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-base
all 2021.20220204-1 [1,128 kB]
Get:51 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libfontbox-java all
1:1.8.16-2 [207 kB]
Get:52 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libpdfbox-java all
1:1.8.16-2 [5,199 kB]
Get:53 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-
recommended all 2021.20220204-1 [14.4 MB]
Get:54 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-pictures
all 2021.20220204-1 [8,720 kB]
Get:55 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-extra
all 2021.20220204-1 [13.9 MB]
Get:56 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-plain-
generic all 2021.20220204-1 [27.5 MB]
Get:57 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tipa all 2:1.3-21
[2,967 kB]
Get:58 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-xetex all
2021.20220204-1 [12.4 MB]
Fetched 202 MB in 12s (17.3 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages ...
Selecting previously unselected package fonts-droid-fallback.
(Reading database ... 123597 files and directories currently installed.)
Preparing to unpack .../00-fonts-droid-fallback_1%3a6.0.1r16-1.1build1_all.deb
...
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1build1) ...
Selecting previously unselected package fonts-lato.
Preparing to unpack .../01-fonts-lato_2.0-2.1_all.deb ...
Unpacking fonts-lato (2.0-2.1) ...
Selecting previously unselected package poppler-data.
Preparing to unpack .../02-poppler-data_0.4.11-1_all.deb ...
Unpacking poppler-data (0.4.11-1) ...
Selecting previously unselected package tex-common.
Preparing to unpack .../03-tex-common_6.17_all.deb ...
Unpacking tex-common (6.17) ...
Selecting previously unselected package fonts-urw-base35.
Preparing to unpack .../04-fonts-urw-base35_20200910-1_all.deb ...
Unpacking fonts-urw-base35 (20200910-1) ...
Selecting previously unselected package libgs9-common.
Preparing to unpack .../05-libgs9-common_9.55.0~dfsg1-0ubuntu5.9_all.deb ...
Unpacking libgs9-common (9.55.0~dfsg1-0ubuntu5.9) ...
Selecting previously unselected package libidn12:amd64.
Preparing to unpack .../06-libidn12_1.38-4ubuntu1_amd64.deb ...
Unpacking libidn12:amd64 (1.38-4ubuntu1) ...

```

```

Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack .../07-libijs-0.35-0.35-15build2_amd64.deb ...
Unpacking libijs-0.35:amd64 (0.35-15build2) ...
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack .../08-libjbig2dec0_0.19-3build2_amd64.deb ...
Unpacking libjbig2dec0:amd64 (0.19-3build2) ...
Selecting previously unselected package libgs9:amd64.
Preparing to unpack .../09-libgs9_9.55.0~dfsg1-0ubuntu5.9_amd64.deb ...
Unpacking libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.9) ...
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack .../10-libkpathsea6_2021.20210626.59705-1ubuntu0.2_amd64.deb
...
Unpacking libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package libwoff1:amd64.
Preparing to unpack .../11-libwoff1_1.0.2-1build4_amd64.deb ...
Unpacking libwoff1:amd64 (1.0.2-1build4) ...
Selecting previously unselected package dvisvgm.
Preparing to unpack .../12-dvisvgm_2.13.1-1_amd64.deb ...
Unpacking dvisvgm (2.13.1-1) ...
Selecting previously unselected package fonts-lmodern.
Preparing to unpack .../13-fonts-lmodern_2.004.5-6.1_all.deb ...
Unpacking fonts-lmodern (2.004.5-6.1) ...
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack .../14-fonts-noto-mono_20201225-1build1_all.deb ...
Unpacking fonts-noto-mono (20201225-1build1) ...
Selecting previously unselected package fonts-texgyre.
Preparing to unpack .../15-fonts-texgyre_20180621-3.1_all.deb ...
Unpacking fonts-texgyre (20180621-3.1) ...
Selecting previously unselected package libapache-pom-java.
Preparing to unpack .../16-libapache-pom-java_18-1_all.deb ...
Unpacking libapache-pom-java (18-1) ...
Selecting previously unselected package libcmark-gfm0.29.0.gfm.3:amd64.
Preparing to unpack .../17-libcmark-gfm0.29.0.gfm.3_0.29.0.gfm.3-3_amd64.deb ...
Unpacking libcmark-gfm0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) ...
Selecting previously unselected package libcmark-gfm-
extensions0.29.0.gfm.3:amd64.
Preparing to unpack .../18-libcmark-gfm-
extensions0.29.0.gfm.3_0.29.0.gfm.3-3_amd64.deb ...
Unpacking libcmark-gfm-extensions0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) ...
Selecting previously unselected package libcommons-parent-java.
Preparing to unpack .../19-libcommons-parent-java_43-1_all.deb ...
Unpacking libcommons-parent-java (43-1) ...
Selecting previously unselected package libcommons-logging-java.
Preparing to unpack .../20-libcommons-logging-java_1.2-2_all.deb ...
Unpacking libcommons-logging-java (1.2-2) ...
Selecting previously unselected package libfontenc1:amd64.
Preparing to unpack .../21-libfontenc1_1%3a1.1.4-1build3_amd64.deb ...
Unpacking libfontenc1:amd64 (1:1.1.4-1build3) ...

```

```

Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack .../22-libptexenc1_2021.20210626.59705-1ubuntu0.2_amd64.deb
...
Unpacking libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package rubygems-integration.
Preparing to unpack .../23-rubygems-integration_1.18_all.deb ...
Unpacking rubygems-integration (1.18) ...
Selecting previously unselected package ruby3.0.
Preparing to unpack .../24-ruby3.0_3.0.2-7ubuntu2.7_amd64.deb ...
Unpacking ruby3.0 (3.0.2-7ubuntu2.7) ...
Selecting previously unselected package ruby-rubygems.
Preparing to unpack .../25-ruby-rubygems_3.3.5-2_all.deb ...
Unpacking ruby-rubygems (3.3.5-2) ...
Selecting previously unselected package ruby.
Preparing to unpack .../26-ruby_1%3a3.0~exp1_amd64.deb ...
Unpacking ruby (1:3.0~exp1) ...
Selecting previously unselected package rake.
Preparing to unpack .../27-rake_13.0.6-2_all.deb ...
Unpacking rake (13.0.6-2) ...
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack .../28-ruby-net-telnet_0.1.1-2_all.deb ...
Unpacking ruby-net-telnet (0.1.1-2) ...
Selecting previously unselected package ruby-webrick.
Preparing to unpack .../29-ruby-webrick_1.7.0-3_all.deb ...
Unpacking ruby-webrick (1.7.0-3) ...
Selecting previously unselected package ruby-xmlrpc.
Preparing to unpack .../30-ruby-xmlrpc_0.3.2-1ubuntu0.1_all.deb ...
Unpacking ruby-xmlrpc (0.3.2-1ubuntu0.1) ...
Selecting previously unselected package libruby3.0:amd64.
Preparing to unpack .../31-libruby3.0_3.0.2-7ubuntu2.7_amd64.deb ...
Unpacking libruby3.0:amd64 (3.0.2-7ubuntu2.7) ...
Selecting previously unselected package libsyntax2:amd64.
Preparing to unpack .../32-libsyntax2_2021.20210626.59705-1ubuntu0.2_amd64.deb
...
Unpacking libsyntax2:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package libteckit0:amd64.
Preparing to unpack .../33-libteckit0_2.5.11+ds1-1_amd64.deb ...
Unpacking libteckit0:amd64 (2.5.11+ds1-1) ...
Selecting previously unselected package libtexlua53:amd64.
Preparing to unpack .../34-libtexlua53_2021.20210626.59705-1ubuntu0.2_amd64.deb
...
Unpacking libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
.../35-libtexluajit2_2021.20210626.59705-1ubuntu0.2_amd64.deb ...
Unpacking libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package libzip-0-13:amd64.
Preparing to unpack .../36-libzip-0-13_0.13.72+dfsg.1-1.1_amd64.deb ...

```

```

Unpacking libzip-0-13:amd64 (0.13.72+dfsg.1-1.1) ...
Selecting previously unselected package xfonts-encodings.
Preparing to unpack .../37-xfonts-encodings_1%3a1.0.5-0ubuntu2_all.deb ...
Unpacking xfonts-encodings (1:1.0.5-0ubuntu2) ...
Selecting previously unselected package xfonts-utils.
Preparing to unpack .../38-xfonts-utils_1%3a7.7+6build2_amd64.deb ...
Unpacking xfonts-utils (1:7.7+6build2) ...
Selecting previously unselected package lmodern.
Preparing to unpack .../39-lmodern_2.004.5-6.1_all.deb ...
Unpacking lmodern (2.004.5-6.1) ...
Selecting previously unselected package pandoc-data.
Preparing to unpack .../40-pandoc-data_2.9.2.1-3ubuntu2_all.deb ...
Unpacking pandoc-data (2.9.2.1-3ubuntu2) ...
Selecting previously unselected package pandoc.
Preparing to unpack .../41-pandoc_2.9.2.1-3ubuntu2_amd64.deb ...
Unpacking pandoc (2.9.2.1-3ubuntu2) ...
Selecting previously unselected package preview-latex-style.
Preparing to unpack .../42-preview-latex-style_12.2-1ubuntu1_all.deb ...
Unpacking preview-latex-style (12.2-1ubuntu1) ...
Selecting previously unselected package t1utils.
Preparing to unpack .../43-t1utils_1.41-4build2_amd64.deb ...
Unpacking t1utils (1.41-4build2) ...
Selecting previously unselected package teckit.
Preparing to unpack .../44-teckit_2.5.11+ds1-1_amd64.deb ...
Unpacking teckit (2.5.11+ds1-1) ...
Selecting previously unselected package tex-gyre.
Preparing to unpack .../45-tex-gyre_20180621-3.1_all.deb ...
Unpacking tex-gyre (20180621-3.1) ...
Selecting previously unselected package texlive-binaries.
Preparing to unpack .../46-texlive-
binaries_2021.20210626.59705-1ubuntu0.2_amd64.deb ...
Unpacking texlive-binaries (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package texlive-base.
Preparing to unpack .../47-texlive-base_2021.20220204-1_all.deb ...
Unpacking texlive-base (2021.20220204-1) ...
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack .../48-texlive-fonts-recommended_2021.20220204-1_all.deb ...
Unpacking texlive-fonts-recommended (2021.20220204-1) ...
Selecting previously unselected package texlive-latex-base.
Preparing to unpack .../49-texlive-latex-base_2021.20220204-1_all.deb ...
Unpacking texlive-latex-base (2021.20220204-1) ...
Selecting previously unselected package libfontbox-java.
Preparing to unpack .../50-libfontbox-java_1%3a1.8.16-2_all.deb ...
Unpacking libfontbox-java (1:1.8.16-2) ...
Selecting previously unselected package libpdfbox-java.
Preparing to unpack .../51-libpdfbox-java_1%3a1.8.16-2_all.deb ...
Unpacking libpdfbox-java (1:1.8.16-2) ...
Selecting previously unselected package texlive-latex-recommended.

```

```

Preparing to unpack .../52-texlive-latex-recommended_2021.20220204-1_all.deb ...
Unpacking texlive-latex-recommended (2021.20220204-1) ...
Selecting previously unselected package texlive-pictures.
Preparing to unpack .../53-texlive-pictures_2021.20220204-1_all.deb ...
Unpacking texlive-pictures (2021.20220204-1) ...
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack .../54-texlive-latex-extra_2021.20220204-1_all.deb ...
Unpacking texlive-latex-extra (2021.20220204-1) ...
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack .../55-texlive-plain-generic_2021.20220204-1_all.deb ...
Unpacking texlive-plain-generic (2021.20220204-1) ...
Selecting previously unselected package tipa.
Preparing to unpack .../56-tipa_2%3a1.3-21_all.deb ...
Unpacking tipa (2:1.3-21) ...
Selecting previously unselected package texlive-xetex.
Preparing to unpack .../57-texlive-xetex_2021.20220204-1_all.deb ...
Unpacking texlive-xetex (2021.20220204-1) ...
Setting up fonts-lato (2.0-2.1) ...
Setting up fonts-noto-mono (20201225-1build1) ...
Setting up libwoff1:amd64 (1.0.2-1build4) ...
Setting up libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Setting up libijs-0.35:amd64 (0.35-15build2) ...
Setting up libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Setting up libfontbox-java (1:1.8.16-2) ...
Setting up rubygems-integration (1.18) ...
Setting up libzip-0-13:amd64 (0.13.72+dfsg.1-1.1) ...
Setting up fonts-urw-base35 (20200910-1) ...
Setting up poppler-data (0.4.11-1) ...
Setting up tex-common (6.17) ...
update-language: texlive-base not installed and configured, doing nothing!
Setting up libfontenc1:amd64 (1:1.1.4-1build3) ...
Setting up libjbig2dec0:amd64 (0.19-3build2) ...
Setting up libteckit0:amd64 (2.5.11+ds1-1) ...
Setting up libapache-pom-java (18-1) ...
Setting up ruby-net-telnet (0.1.1-2) ...
Setting up xfonts-encodings (1:1.0.5-0ubuntu2) ...
Setting up t1utils (1.41-4build2) ...
Setting up libidn12:amd64 (1.38-4ubuntu1) ...
Setting up fonts-texgyre (20180621-3.1) ...
Setting up libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Setting up ruby-webrick (1.7.0-3) ...
Setting up libcmark-gfm0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) ...
Setting up fonts-lmodern (2.004.5-6.1) ...
Setting up libcmark-gfm-extensions0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) ...
Setting up fonts-droid-fallback (1:6.0.1r16-1.1build1) ...
Setting up pandoc-data (2.9.2.1-3ubuntu2) ...
Setting up ruby-xmlrpc (0.3.2-1ubuntu0.1) ...
Setting up libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) ...

```

```

Setting up libgs9-common (9.55.0~dfsg1-0ubuntu5.9) ...
Setting up teckit (2.5.11+ds1-1) ...
Setting up libpdfbox-java (1:1.8.16-2) ...
Setting up libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.9) ...
Setting up preview-latex-style (12.2-1ubuntu1) ...
Setting up libcommons-parent-java (43-1) ...
Setting up dvisvgm (2.13.1-1) ...
Setting up libcommons-logging-java (1.2-2) ...
Setting up xfonts-utils (1:7.7+6build2) ...
Setting up libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Setting up pandoc (2.9.2.1-3ubuntu2) ...
Setting up texlive-binaries (2021.20210626.59705-1ubuntu0.2) ...
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up lmodern (2.004.5-6.1) ...
Setting up texlive-base (2021.20220204-1) ...
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST...
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN...
mktexlsr: Updating /var/lib/texmf/ls-R...
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4: /var/lib/texmf/tex/generic/tex-
ini-files/pdftexconfig.tex
Setting up tex-gyre (20180621-3.1) ...
Setting up texlive-plain-generic (2021.20220204-1) ...
Setting up texlive-latex-base (2021.20220204-1) ...
Setting up texlive-latex-recommended (2021.20220204-1) ...
Setting up texlive-pictures (2021.20220204-1) ...
Setting up texlive-fonts-recommended (2021.20220204-1) ...
Setting up tipa (2:1.3-21) ...
Setting up texlive-latex-extra (2021.20220204-1) ...
Setting up texlive-xetex (2021.20220204-1) ...
Setting up rake (13.0.6-2) ...
Setting up libruby3.0:amd64 (3.0.2-7ubuntu2.7) ...
Setting up ruby3.0 (3.0.2-7ubuntu2.7) ...
Setting up ruby (1:3.0~exp1) ...
Setting up ruby-rubygems (3.3.5-2) ...
Processing triggers for man-db (2.10.2-1) ...

```

```
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) ...
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero.so.0 is not a
symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libur_loader.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_opencl.so.0 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

Processing triggers for tex-common (6.17) ...
Running updmap-sys. This may take some time... done.
Running mktexlsr /var/lib/texmf ... done.
Building format(s) --all.
    This may take some time...
```