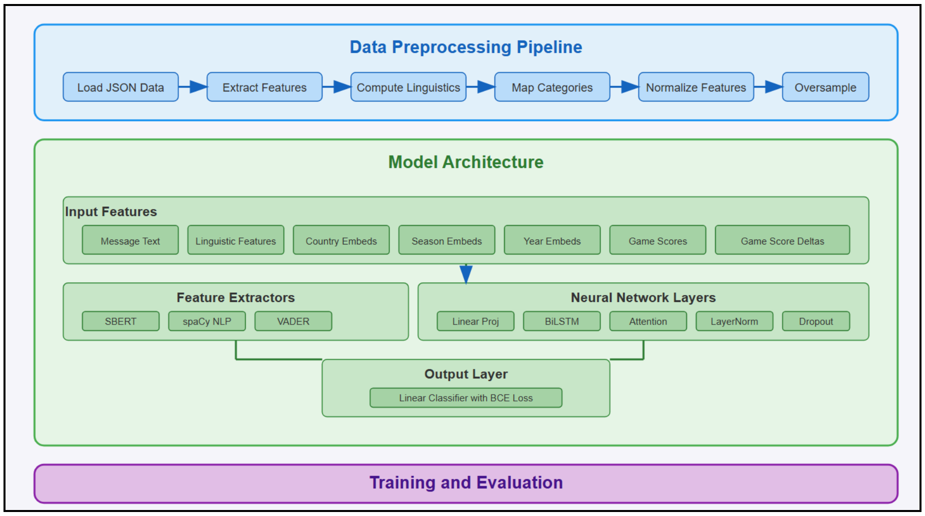
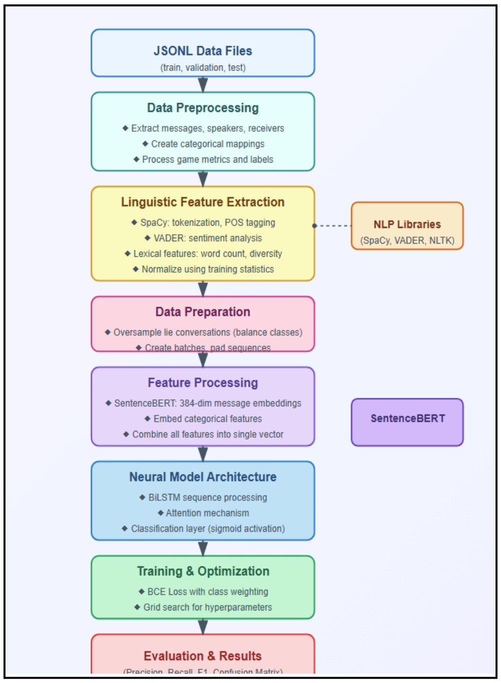
All Model Results

1. Self attention - Bidirectional LSTM + Oversampling

CODE LINK

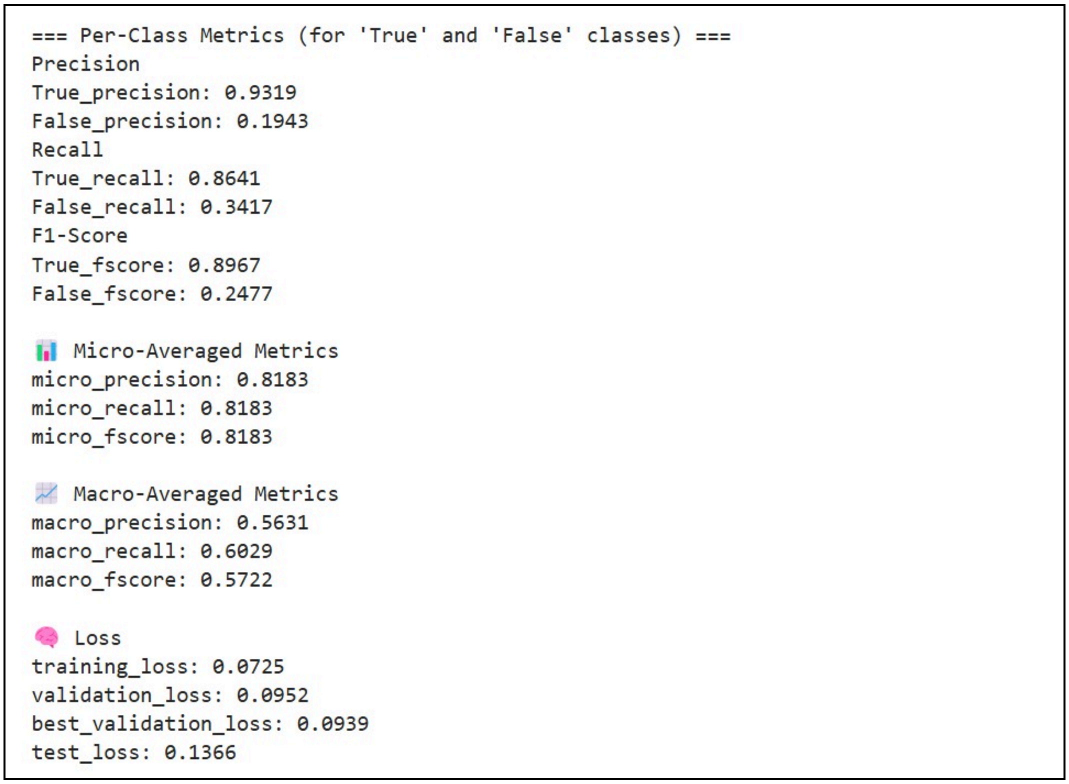



JSONL Data Files
(train, validation, test)
Data Preprocessing
◆ Extract messages, speakers, receivers
◆ Create categorical mappings
◆ Process game metrics and labels
Linguistic Feature Extraction
◆ SpaCy: tokenization, POS tagging
◆ VADER: sentiment analysis
◆ Lexical features: word count, diversity
◆ Normalize using training statistics
NLP Libraries
(SpaCy, VADER, NLTK)
Data Preparation
◆ Oversample lie conversations (balance classes)
◆ Create batches, pad sequences
Feature Processing
◆ SentenceBERT: 384-dim message embeddings
◆ Embed categorical features
◆ Combine all features into single vector
SentenceBERT
Neural Model Architecture
◆ BiLSTM sequence processing
◆ Attention mechanism
◆ Classification layer (sigmoid activation)
Training & Optimization
◆ BCE Loss with class weighting
◆ Grid search for hyperparameters
Evaluation & Results
(Precision, Recall, F1, Confusion Matrix)

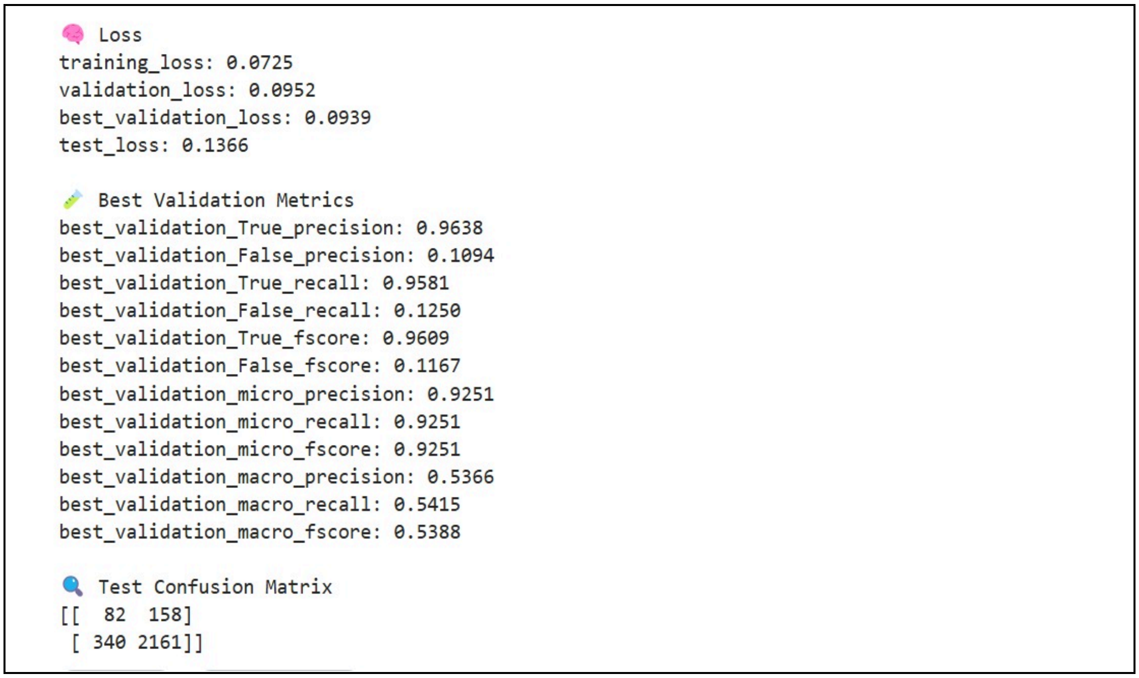
[Start]
|
v
[Install Libraries]
- Install sentence-transformers, spacy, nltk
- Download spaCy model and VADER lexicon

```
 |
 v
[Setup Environment]
 - Configure logging
 - Set device (GPU/CPU)
 - Load Sentence-BERT, spaCy, VADER
 |
 v
[EnhancedDeceptionDetector]
 - Initialize embeddings (country, season, year)
 - Define LSTM, attention, classifier
 - Implement forward pass
 |
 v
[Preprocess Data]
 - Load train.jsonl
 - Create country, season, year mappings
 - Extract features (messages, linguistic, categorical, numerical)
 - Normalize linguistic features
 - Repeat for val.jsonl, test.jsonl using train mappings
 |
 v
[Oversample Lies]
 - Duplicate conversations with lies (factor=2)
 |
 v
[Grid Search Hyperparameters]
 - For each lr, batch_size combination:
   |
   v
   [Train Model]
     - Initialize model
     - Train with Adam, warmup scheduler
     - Use BCEWithLogitsLoss with pos_weight
```

```
      - Apply early stopping
      |
      v
    [Evaluate on Validation]
      - Compute loss, metrics (precision, recall, F1)
      - Save best model based on validation loss
      |
      v
    [Evaluate on Test]
      - Load best model
      - Compute test loss, metrics
     |
     v
    [Track Best Parameters]
      - Update if validation macro F1 improves
  |
  v
[Output Results]
  - Print best parameters
  - Display per-class metrics (True/False)
  - Show micro/macro metrics
  - Report losses
  - Present confusion matrix
  |
  v
[End]
```



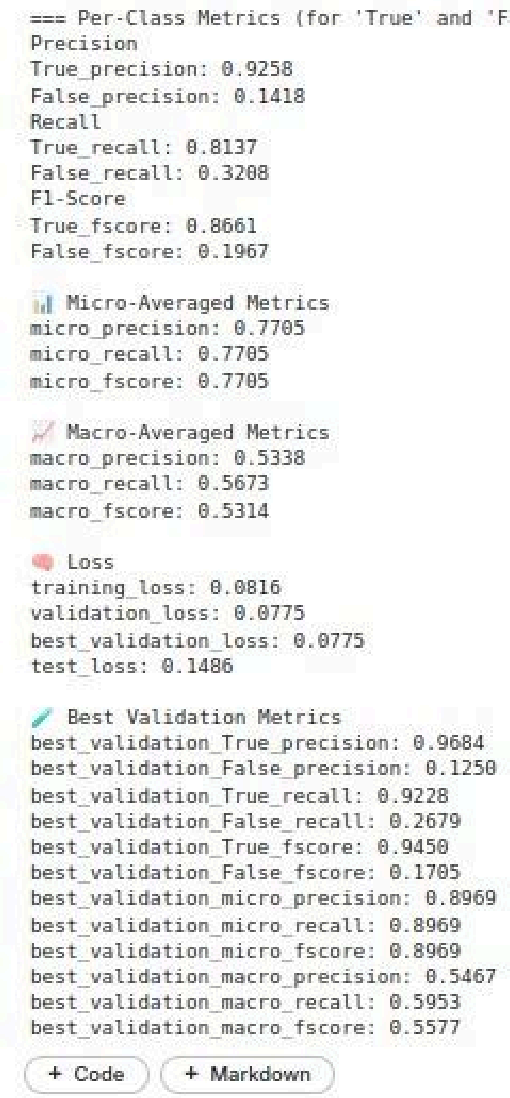
```
=== Per-Class Metrics (for 'True' and 'False' classes) ===
Precision
True_precision: 0.9319
False_precision: 0.1943
Recall
True_recall: 0.8641
False_recall: 0.3417
F1-Score
True_fscore: 0.8967
False_fscore: 0.2477

📊 Micro-Averaged Metrics
micro_precision: 0.8183
micro_recall: 0.8183
micro_fscore: 0.8183

📈 Macro-Averaged Metrics
macro_precision: 0.5631
macro_recall: 0.6029
macro_fscore: 0.5722

🧠 Loss
training_loss: 0.0725
validation_loss: 0.0952
best_validation_loss: 0.0939
test_loss: 0.1366
```



```
🧠 Loss
training_loss: 0.0725
validation_loss: 0.0952
best_validation_loss: 0.0939
test_loss: 0.1366

🧪 Best Validation Metrics
best_validation_True_precision: 0.9638
best_validation_False_precision: 0.1094
best_validation_True_recall: 0.9581
best_validation_False_recall: 0.1250
best_validation_True_fscore: 0.9609
best_validation_False_fscore: 0.1167
best_validation_micro_precision: 0.9251
best_validation_micro_recall: 0.9251
best_validation_micro_fscore: 0.9251
best_validation_macro_precision: 0.5366
best_validation_macro_recall: 0.5415
best_validation_macro_fscore: 0.5388

🔍 Test Confusion Matrix
[[  82  158]
 [ 340 2161]]
```

Simple Lstm + All Features

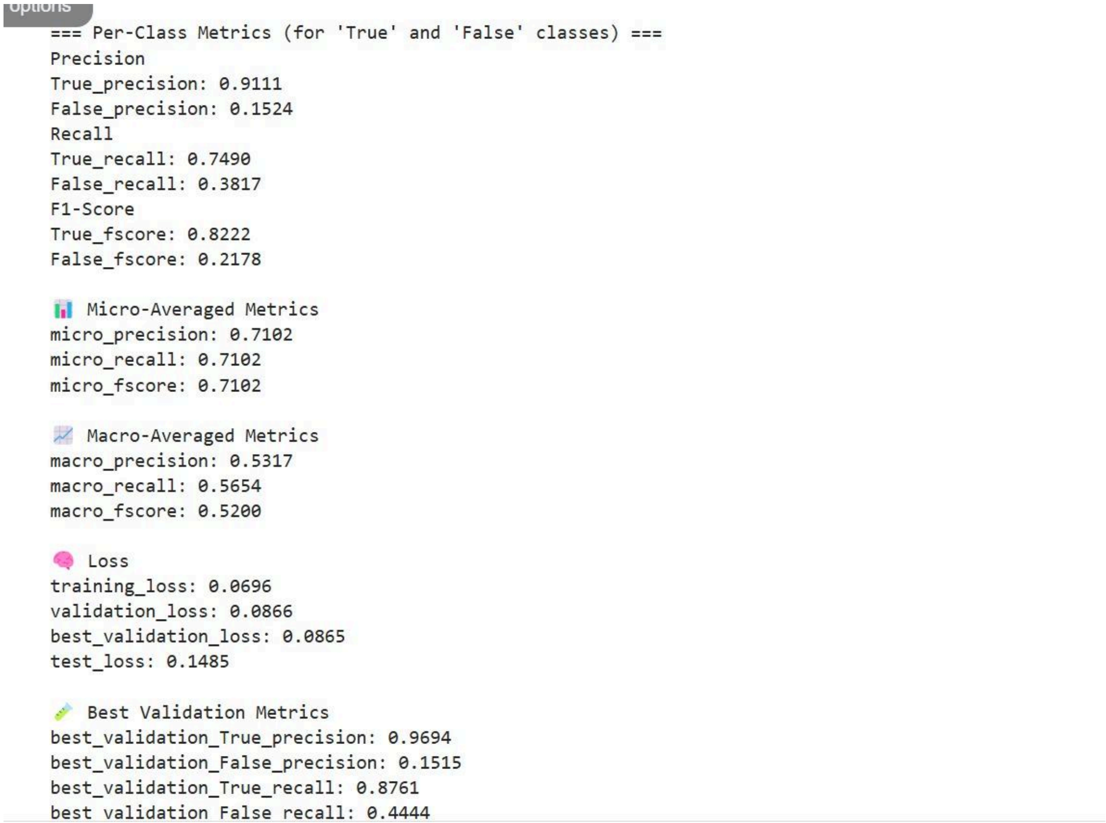
Code-https://github.com/palak-b19/Deception-Detection/blob/main/Final_Codes/simple_lstm_all_features.py



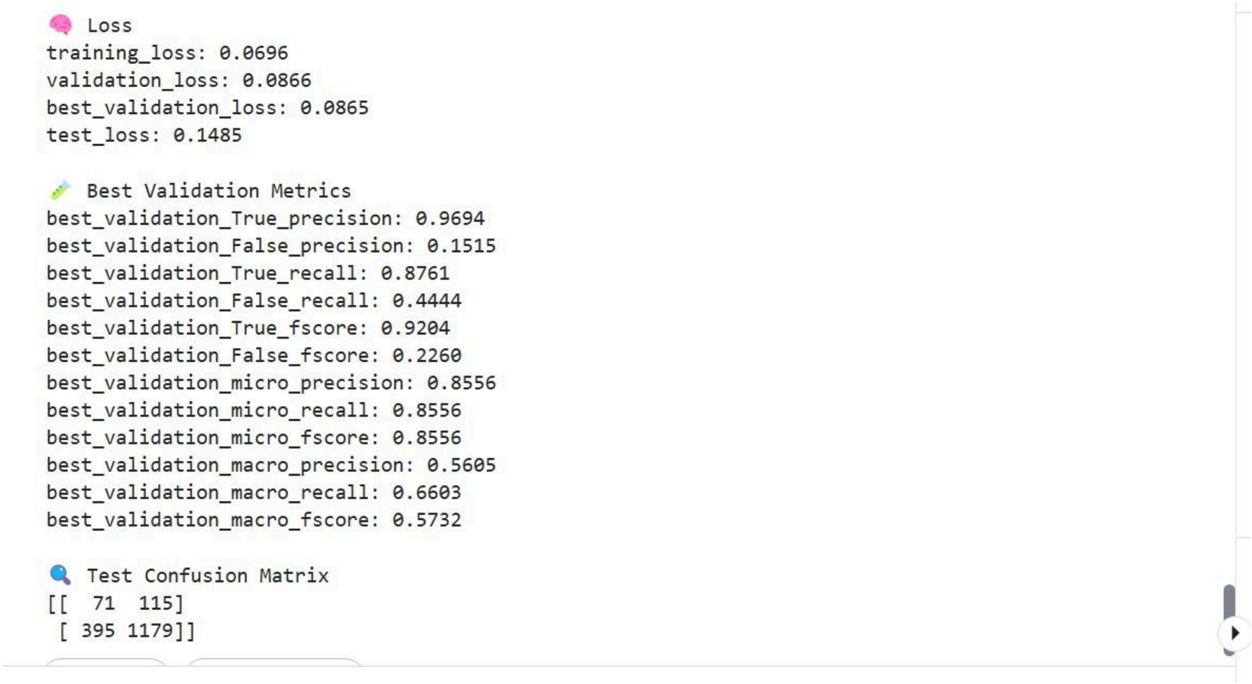
```
=== Per-Class Metrics (for 'True' and 'False' classes) ===
Precision
True_precision: 0.9258
False_precision: 0.1418
Recall
True_recall: 0.8137
False_recall: 0.3208
F1-Score
True_fscore: 0.8661
False_fscore: 0.1967

📊 Micro-Averaged Metrics
micro_precision: 0.7705
micro_recall: 0.7705
micro_fscore: 0.7705

📈 Macro-Averaged Metrics
macro_precision: 0.5338
macro_recall: 0.5673
macro_fscore: 0.5314

🧠 Loss
training_loss: 0.0816
validation_loss: 0.0775
best_validation_loss: 0.0775
test_loss: 0.1486

🧪 Best Validation Metrics
best_validation_True_precision: 0.9684
best_validation_False_precision: 0.1250
best_validation_True_recall: 0.9228
best_validation_False_recall: 0.2679
best_validation_True_fscore: 0.9450
best_validation_False_fscore: 0.1705
best_validation_micro_precision: 0.8969
best_validation_micro_recall: 0.8969
best_validation_micro_fscore: 0.8969
best_validation_macro_precision: 0.5467
best_validation_macro_recall: 0.5953
best_validation_macro_fscore: 0.5577
```

Self attention + Positional Encoding - Bidirectional LSTM + Oversampling

Code - **CODE LINK**



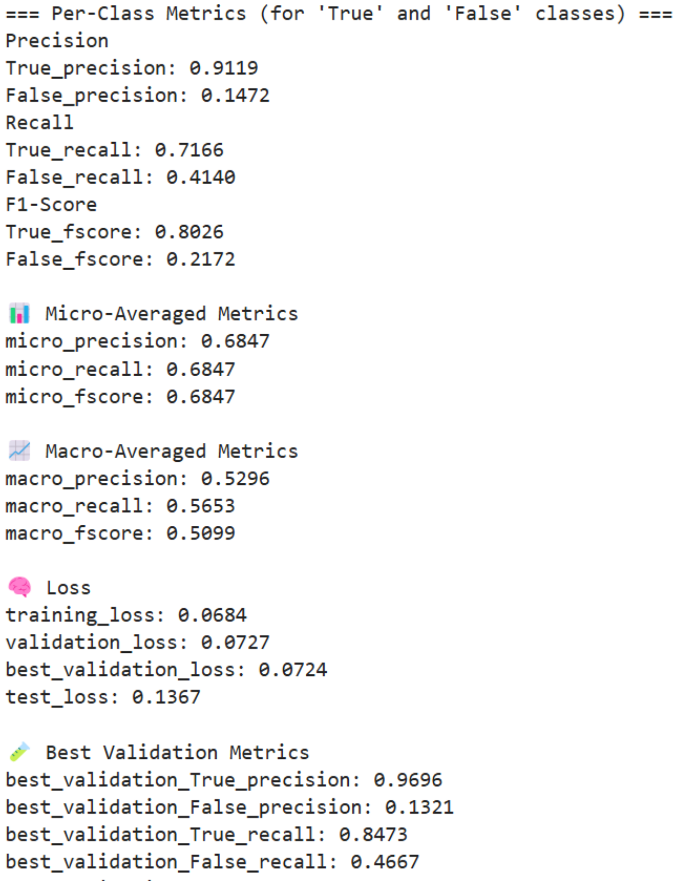
```
=== Per-Class Metrics (for 'True' and 'False' classes) ===
Precision
True_precision: 0.9111
False_precision: 0.1524
Recall
True_recall: 0.7490
False_recall: 0.3817
F1-Score
True_fscore: 0.8222
False_fscore: 0.2178

📊 Micro-Averaged Metrics
micro_precision: 0.7102
micro_recall: 0.7102
micro_fscore: 0.7102

📈 Macro-Averaged Metrics
macro_precision: 0.5317
macro_recall: 0.5654
macro_fscore: 0.5200

🧠 Loss
training_loss: 0.0696
validation_loss: 0.0866
best_validation_loss: 0.0865
test_loss: 0.1485

🧪 Best Validation Metrics
best_validation_True_precision: 0.9694
best_validation_False_precision: 0.1515
best_validation_True_recall: 0.8761
best_validation_False_recall: 0.4444
```



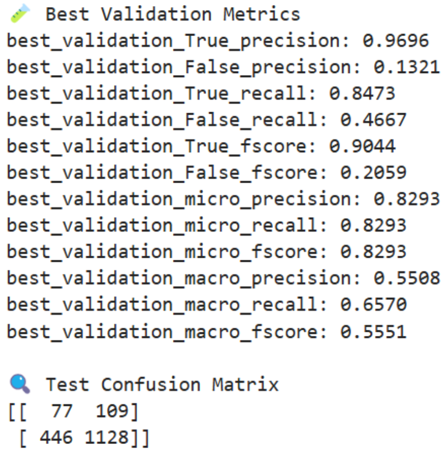
```
🧠 Loss
training_loss: 0.0696
validation_loss: 0.0866
best_validation_loss: 0.0865
test_loss: 0.1485

🧪 Best Validation Metrics
best_validation_True_precision: 0.9694
best_validation_False_precision: 0.1515
best_validation_True_recall: 0.8761
best_validation_False_recall: 0.4444
```

```
best_validation_True_fscore: 0.9204
best_validation_False_fscore: 0.2260
best_validation_micro_precision: 0.8556
best_validation_micro_recall: 0.8556
best_validation_micro_fscore: 0.8556
best_validation_macro_precision: 0.5605
best_validation_macro_recall: 0.6603
best_validation_macro_fscore: 0.5732

🔍 Test Confusion Matrix
[[  71  115]
 [ 395 1179]]
```

Bidirectional LSTM + Primarily Oversampling



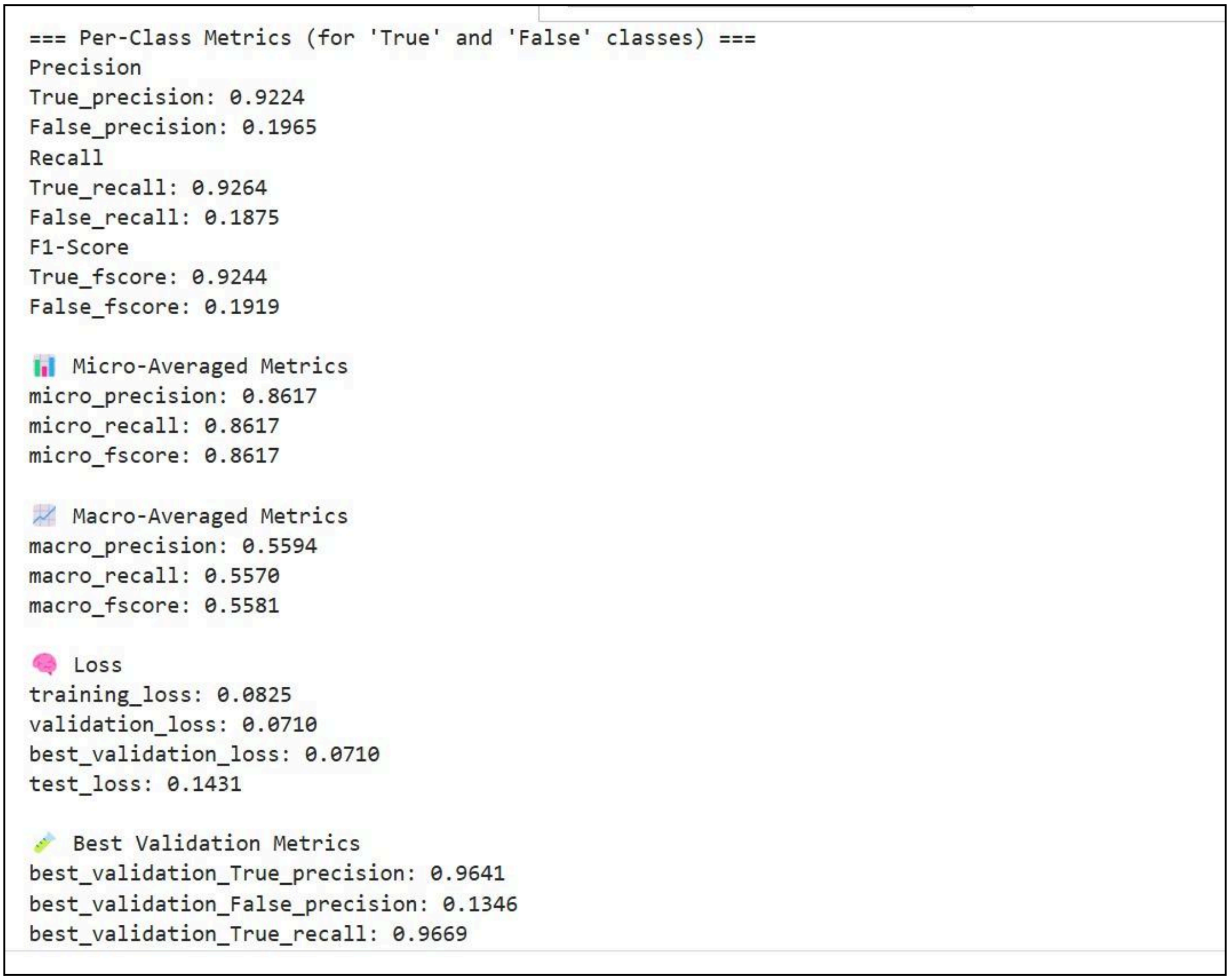
```
=== Per-Class Metrics (for 'True' and 'False' classes) ===
Precision
True_precision: 0.9119
False_precision: 0.1472
Recall
True_recall: 0.7166
False_recall: 0.4140
F1-Score
True_fscore: 0.8026
False_fscore: 0.2172

📊 Micro-Averaged Metrics
micro_precision: 0.6847
micro_recall: 0.6847
micro_fscore: 0.6847

📈 Macro-Averaged Metrics
macro_precision: 0.5296
macro_recall: 0.5653
macro_fscore: 0.5099

🧠 Loss
training_loss: 0.0684
validation_loss: 0.0727
best_validation_loss: 0.0724
test_loss: 0.1367

🧪 Best Validation Metrics
best_validation_True_precision: 0.9696
best_validation_False_precision: 0.1321
best_validation_True_recall: 0.8473
best_validation_False_recall: 0.4667
```



```
🧪 Best Validation Metrics
best_validation_True_precision: 0.9696
best_validation_False_precision: 0.1321
best_validation_True_recall: 0.8473
best_validation_False_recall: 0.4667
best_validation_True_fscore: 0.9044
best_validation_False_fscore: 0.2059
best_validation_micro_precision: 0.8293
best_validation_micro_recall: 0.8293
best_validation_micro_fscore: 0.8293
best_validation_macro_precision: 0.5508
best_validation_macro_recall: 0.6570
best_validation_macro_fscore: 0.5551

🔍 Test Confusion Matrix
[[  77  109]
 [ 446 1128]]
```

Multiheaded Attention - Bidirectional LSTM + Oversampling

Code - CODE LINK



```
=== Per-Class Metrics (for 'True' and 'False' classes) ===
Precision
True_precision: 0.9224
False_precision: 0.1965
Recall
True_recall: 0.9264
False_recall: 0.1875
F1-Score
True_fscore: 0.9244
False_fscore: 0.1919

📊 Micro-Averaged Metrics
micro_precision: 0.8617
micro_recall: 0.8617
micro_fscore: 0.8617

📈 Macro-Averaged Metrics
macro_precision: 0.5594
macro_recall: 0.5570
macro_fscore: 0.5581

🧠 Loss
training_loss: 0.0825
validation_loss: 0.0710
best_validation_loss: 0.0710
test_loss: 0.1431

🧪 Best Validation Metrics
best_validation_True_precision: 0.9641
best_validation_False_precision: 0.1346
best_validation_True_recall: 0.9669
```

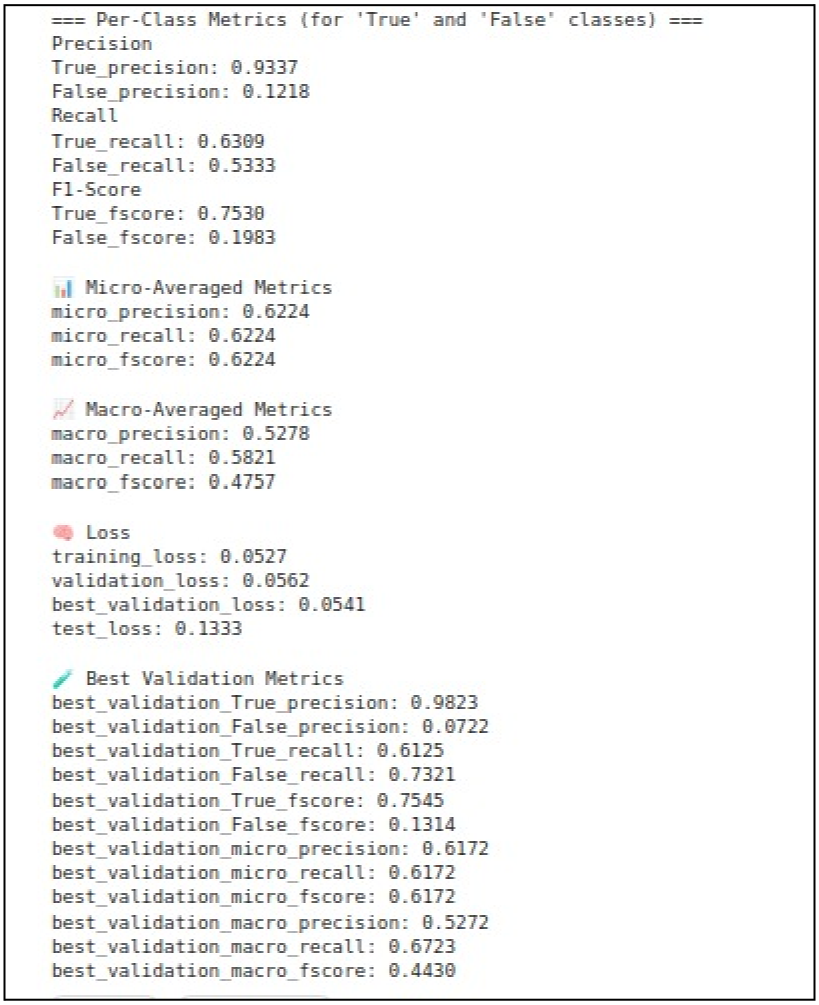
Dual LSTM

Conversation Sequence: [Message1, Message2, ..., MessageN]
For each Message i: {text_i, sender_i, receiver_i, sender_label_i, receiver_label_i}

1. Preprocessing:
   - text_i → Message Embedding (e.g., via word embeddings or BERT)

```
  - sender_i → Sender Embedding (e.g., one-hot or learned embedding for the country)
  - receiver_i → Receiver Embedding (e.g., one-hot or learned embedding for the country)

2. Sender LSTM:
  - Input at step i: [Message Embedding_i, Sender Embedding_i]
  - Output: Hidden State h_sender_i
  - Prediction: sender_pred_i (probability of sender_label_i)

3. Receiver LSTM:
  - Input at step i: [Message Embedding_i, Receiver Embedding_i]
  - Output: Hidden State h_receiver_i
  - Prediction: receiver_pred_i (probability of receiver_label_i, if annotated)

4. Interaction Layer:
  - Combine: h_combined_i = concatenate(h_sender_i, h_receiver_i)
  - Classifier: h_combined_i → Final Prediction (probability of sender_label_i)

5. Training:
  - Loss_Sender = CrossEntropy(sender_pred_i, sender_label_i)
  - Loss_Receiver = CrossEntropy(receiver_pred_i, receiver_label_i) [if available, else masked]
  - Loss_Combined = CrossEntropy(final_prediction_i, sender_label_i)
  - Total Loss = w1 * Loss_Sender + w2 * Loss_Receiver + w3 * Loss_Combined
  - Optimize model parameters

6. For Each Conversation:
  - Reset LSTM hidden states
  - Process sequence independently
  - Repeat for next pair (e.g., Austria-Italy → England-France)
```

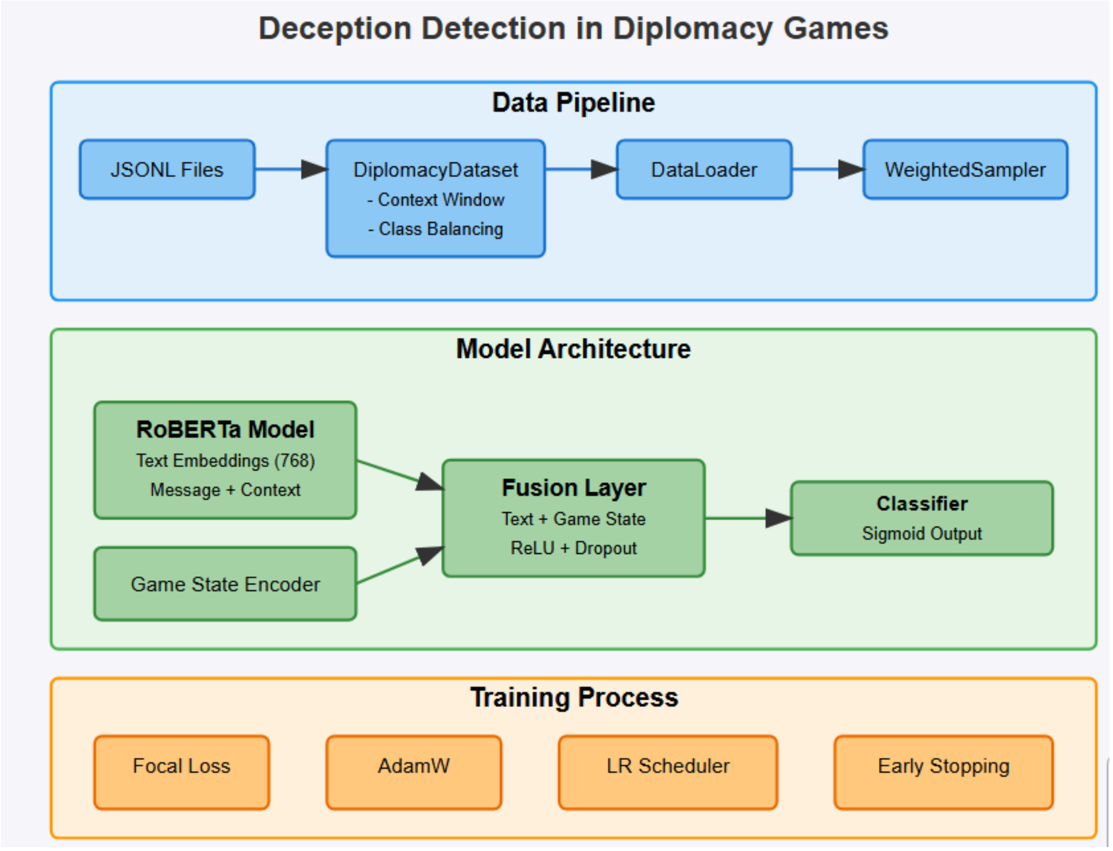
Code- CODE LINK



```
=== Per-Class Metrics (for 'True' and 'False' classes) ===
Precision
True_precision: 0.9337
False_precision: 0.1218
Recall
True_recall: 0.6309
False_recall: 0.5333
F1-Score
True_fscore: 0.7530
False_fscore: 0.1983

📊 Micro-Averaged Metrics
micro_precision: 0.6224
micro_recall: 0.6224
micro_fscore: 0.6224

📈 Macro-Averaged Metrics
macro_precision: 0.5278
macro_recall: 0.5821
macro_fscore: 0.4757

🧠 Loss
training_loss: 0.0527
validation_loss: 0.0562
best_validation_loss: 0.0541
test_loss: 0.1333

🧪 Best Validation Metrics
best_validation_True_precision: 0.9823
best_validation_False_precision: 0.0722
best_validation_True_recall: 0.6125
best_validation_False_recall: 0.7321
best_validation_True_fscore: 0.7545
best_validation_False_fscore: 0.1314
best_validation_micro_precision: 0.6172
best_validation_micro_recall: 0.6172
best_validation_micro_fscore: 0.6172
best_validation_macro_precision: 0.5272
best_validation_macro_recall: 0.6723
best_validation_macro_fscore: 0.4430
```

Roberta

Code-CODE LINK



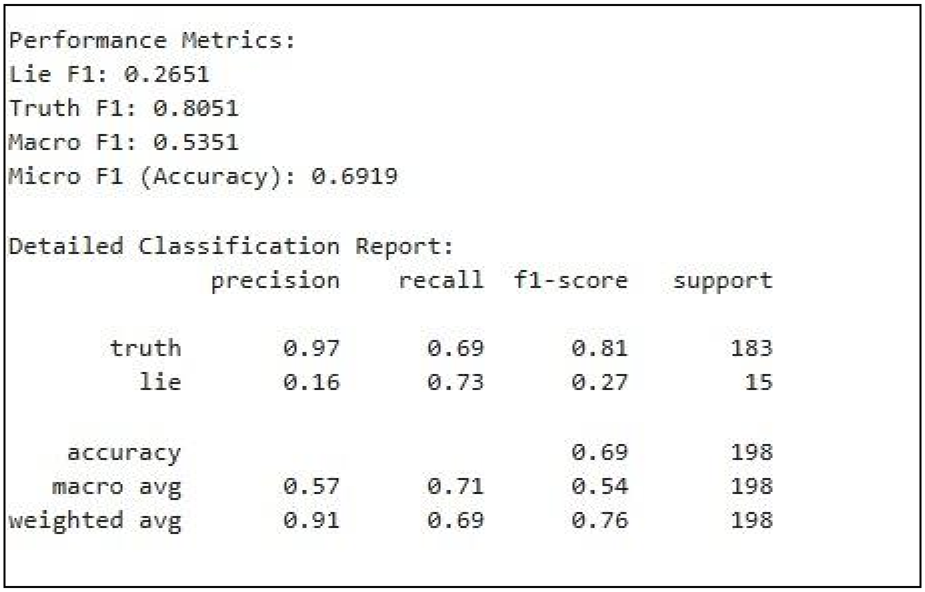
```
Epoch 1/15, Loss: 0.1798, Macro F1: 0.5026, Lie F1: 0.0945, Accuracy: 0.8376
New best model saved with Lie F1: 0.0945
Evaluation  batches:  100%
  45/45  [00:04<00:00,    9.29it/s]
Epoch 2/15, Loss: 0.1657, Macro F1: 0.4803, Lie F1: 0.1089, Accuracy: 0.7458
New best model saved with Lie F1: 0.1089
Evaluation  batches:  100%
  45/45  [00:04<00:00,    9.31it/s]
Epoch 3/15, Loss: 0.0876, Macro F1: 0.5196, Lie F1: 0.0863, Accuracy: 0.9103
Evaluation  batches:  100%
  45/45  [00:04<00:00,    9.31it/s]
Epoch 4/15, Loss: 0.0560, Macro F1: 0.5191, Lie F1: 0.0682, Accuracy: 0.9421
Evaluation  batches:  100%
  45/45  [00:04<00:00,    9.28it/s]
Epoch 5/15, Loss: 0.0387, Macro F1: 0.5332, Lie F1: 0.1043, Accuracy: 0.9273
Evaluation  batches:  100%
```

```
  45/45  [00:04<00:00,    9.33it/s]
Epoch 6/15, Loss: 0.0375, Macro F1: 0.5364, Lie F1: 0.1042, Accuracy: 0.9393
Evaluation  batches:  100%
  45/45  [00:04<00:00,    9.24it/s]
Epoch 7/15, Loss: 0.0268, Macro F1: 0.5111, Lie F1: 0.0709, Accuracy: 0.9075
Evaluation  batches:  100%
  45/45  [00:04<00:00,    9.25it/s]
Epoch 8/15, Loss: 0.0160, Macro F1: 0.5236, Lie F1: 0.0862, Accuracy: 0.9251
Evaluation  batches:  100%
  45/45  [00:04<00:00,    9.29it/s]
Epoch 9/15, Loss: 0.0095, Macro F1: 0.5133, Lie F1: 0.0526, Accuracy: 0.9492
Evaluation  batches:  100%
  45/45  [00:04<00:00,    9.30it/s]
Epoch 10/15, Loss: 0.0082, Macro F1: 0.5107, Lie F1: 0.0494, Accuracy: 0.9456
Evaluation  batches:  100%
  45/45  [00:04<00:00,    9.30it/s]
Epoch 11/15, Loss: 0.0045, Macro F1: 0.5044, Lie F1: 0.0312, Accuracy: 0.9562
Evaluation  batches:  100%
  45/45  [00:04<00:00,    9.28it/s]
Epoch 12/15, Loss: 0.0039, Macro F1: 0.5143, Lie F1: 0.0541, Accuracy: 0.9506
```

Test Macro F1: 0.4885, Lie F1: 0.1595, Accuracy: 0.7001

LLM feedback loop

Code-CODE LINK



```
Performance Metrics:
Lie F1: 0.2651
Truth F1: 0.8051
Macro F1: 0.5351
Micro F1 (Accuracy): 0.6919

Detailed Classification Report:
              precision    recall  f1-score   support

       truth       0.97      0.69      0.81       183
         lie       0.16      0.73      0.27        15

    accuracy                           0.69       198
   macro avg       0.57      0.71      0.54       198
weighted avg       0.91      0.69      0.76       198
```