# BasicLSTM

root@autodl-container-bcde4980ca-ef9f2701:~/hate/hate-speech-detection-survey# python /root/hate/hate-speech-detection-survey/src/training/main.py

[nltk\_data] Downloading package stopwords to /root/nltk\_data...

[nltk\_data] Package stopwords is already up-to-date!

Device: cuda

file loaded and formatted..

data split into train/val/test

field objects created

fields and dataset object created

vocabulary built..

dataloaders created..

model\_type: BasicLSTM

optimizer\_type: adam

loss\_criterion: bcelosswithlogits

learning rate: 0.001

epochs: 10

patience\_es: 5

scheduler\_type:

patience\_lr: 5

save\_condition: acc

Model BasicLSTM loaded on cuda

Loss used: BCEWithLogitsLoss()

Optimizer used: Adam (

Parameter Group 0

amsgrad: False

betas: (0.9, 0.999)

eps: 1e-08

lr: 0.001

weight\_decay: 0

)

Scheduler used: None

\*\*TRAINING\*\*

Epoch 0/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|█████████████████████████████████████████████████████████████████████| 83/83 [00:00<00:00, 106.07it/s]

train Loss: 0.6124 Acc: 0.4103

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 282.41it/s]

val Loss: 0.6050 Acc: 0.5097

Epoch complete in 0.9s

Epoch 1/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|█████████████████████████████████████████████████████████████████████| 83/83 [00:00<00:00, 117.93it/s]

train Loss: 0.5141 Acc: 0.6397

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 283.90it/s]

val Loss: 0.5662 Acc: 0.6304

Epoch complete in 0.8s

Epoch 2/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|█████████████████████████████████████████████████████████████████████| 83/83 [00:00<00:00, 108.73it/s]

train Loss: 0.3930 Acc: 0.7813

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 265.28it/s]

val Loss: 0.5947 Acc: 0.6620

Epoch complete in 0.9s

Epoch 3/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|█████████████████████████████████████████████████████████████████████| 83/83 [00:00<00:00, 100.88it/s]

train Loss: 0.2660 Acc: 0.8732

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 266.78it/s]

val Loss: 0.7423 Acc: 0.6565

Epoch complete in 0.9s

EarlyStopping counter: 1 out of 5

Epoch 4/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|█████████████████████████████████████████████████████████████████████| 83/83 [00:00<00:00, 112.71it/s]

train Loss: 0.1597 Acc: 0.9301

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 281.35it/s]

val Loss: 0.8181 Acc: 0.6765

Epoch complete in 0.8s

Epoch 5/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|█████████████████████████████████████████████████████████████████████| 83/83 [00:00<00:00, 107.52it/s]

train Loss: 0.0947 Acc: 0.9630

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 205.52it/s]

val Loss: 1.0192 Acc: 0.6656

Epoch complete in 0.9s

EarlyStopping counter: 1 out of 5

Epoch 6/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:00<00:00, 93.02it/s]

train Loss: 0.0623 Acc: 0.9766

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 286.17it/s]

val Loss: 1.0826 Acc: 0.6886

Epoch complete in 1.0s

Epoch 7/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:00<00:00, 98.35it/s]

train Loss: 0.0458 Acc: 0.9817

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 278.17it/s]

val Loss: 1.2502 Acc: 0.6684

Epoch complete in 0.9s

EarlyStopping counter: 1 out of 5

Epoch 8/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|█████████████████████████████████████████████████████████████████████| 83/83 [00:00<00:00, 108.16it/s]

train Loss: 0.0300 Acc: 0.9896

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 273.91it/s]

val Loss: 1.3822 Acc: 0.6858

Epoch complete in 0.9s

EarlyStopping counter: 2 out of 5

Epoch 9/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|█████████████████████████████████████████████████████████████████████| 83/83 [00:00<00:00, 108.70it/s]

train Loss: 0.0259 Acc: 0.9907

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 292.77it/s]

val Loss: 1.4535 Acc: 0.6856

Epoch complete in 0.8s

EarlyStopping counter: 3 out of 5

Training complete in 0m 9s

Best val Acc: 0.688600

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████████████████████████████████████████████████████████████████| 860/860 [00:00<00:00, 995.98it/s]

Test stats - Loss: 0.9920 Acc: 69.25%

Inference on Testset complete in 0.9s

Model saved at saved-models/BasicLSTM\_2025-04-26\_20-47-46\_trained\_testAcc=0.6925.pth

Training graph saved at figures/BasicLSTM\_losses&acc\_2025-04-26\_20-47-46\_testAcc=0.6925.png

Confusion Matrix saved at figures/BasicLSTM\_CM\_2025-04-26\_20-47-46\_testAcc=0.6925.png

Accuracy: 76.512%

Macro F1-score: 69.246%

Binary F1-score: 54.299%

MSE: 0.235

precision recall f1-score support

class 0 0.82 0.87 0.84 620

class 1 0.59 0.50 0.54 240

accuracy 0.77 860

macro avg 0.71 0.68 0.69 860

weighted avg 0.76 0.77 0.76 860

root@autodl-container-bcde4980ca-ef9f2701:~/hate/hate-speech-detection-survey# 这个结果说明了啥

从你给出的训练和测试输出结果，可以从以下几个方面进行分析解读：

### 1. 训练过程

- \*\*训练损失和准确率\*\*：在训练过程中，训练损失（`train Loss`）随着训练轮次（`Epoch`）的增加而持续下降，从初始的 0.6124 降到了 0.0259；训练准确率（`train Acc`）则不断上升，从 0.4103 提升到了 0.9907。这表明模型在训练数据上的学习效果良好，能够逐渐拟合训练数据。

- \*\*验证损失和准确率\*\*：验证损失（`val Loss`）在前期有一定波动，之后开始上升，从 0.6050 上升到了 1.4535；验证准确率（`val Acc`）虽然有一定起伏，但整体提升幅度不大，最高达到 0.6886。这可能暗示模型出现了过拟合现象，即模型在训练数据上表现很好，但在未见过的验证数据上泛化能力不足。

- \*\*早停机制\*\*：代码中设置了早停机制（`EarlyStopping`），当验证准确率没有明显提升时，计数器会增加。当计数器达到预设的耐心值（`patience\_es = 5`）时，训练会提前停止。从输出结果来看，早停机制在一定程度上发挥了作用，避免了模型过度训练。

### 2. 测试结果

- \*\*测试损失和准确率\*\*：测试损失为 0.9920，测试准确率为 69.25%。这表明模型在测试数据上的表现一般，与训练准确率相比有较大差距，进一步验证了模型可能存在过拟合问题。

- \*\*模型评估指标\*\*：除了准确率，还给出了宏 F1 分数（Macro F1-score）、二分类 F1 分数（Binary F1-score）和均方误差（MSE）等评估指标。宏 F1 分数为 69.246%，二分类 F1 分数为 54.299%，MSE 为 0.235。这些指标可以更全面地评估模型的性能，其中 F1 分数综合考虑了精确率和召回率，MSE 则衡量了模型预测值与真实值之间的平均误差。

- \*\*混淆矩阵\*\*：输出结果中还给出了混淆矩阵的详细信息，包括不同类别的精确率（`precision`）、召回率（`recall`）和 F1 分数。从混淆矩阵可以看出，模型在类别 0 上的表现较好，精确率为 0.82，召回率为 0.87；而在类别 1 上的表现相对较差，精确率为 0.59，召回率为 0.50。这可能意味着模型在识别类别 1 时存在一定困难，需要进一步优化。

### 3. 模型保存

- \*\*模型文件\*\*：训练好的模型被保存到了 `saved-models/BasicLSTM\_2025-04-26\_20-47-46\_trained\_testAcc=0.6925.pth` 文件中，方便后续使用。

- \*\*训练可视化\*\*：训练过程中的损失和准确率曲线被保存到了 `figures/BasicLSTM\_losses&acc\_2025-04-26\_20-47-46\_testAcc=0.6925.png` 文件中，混淆矩阵被保存到了 `figures/BasicLSTM\_CM\_2025-04-26\_20-47-46\_testAcc=0.6925.png` 文件中，这些可视化结果有助于进一步分析模型的训练过程和性能。

### 改进建议

- \*\*数据增强\*\*：可以尝试使用数据增强技术，如随机替换、插入、删除等操作，增加训练数据的多样性，提高模型的泛化能力。

- \*\*正则化\*\*：在模型训练过程中，可以添加正则化项，如 L1 或 L2 正则化，以减少模型的过拟合。

- \*\*调整模型结构\*\*：可以尝试调整模型的结构，如增加或减少层数、调整隐藏单元的数量等，以找到更合适的模型架构。

- \*\*调整超参数\*\*：可以使用网格搜索或随机搜索等方法，调整学习率、批次大小、训练轮数等超参数，以优化模型的性能。

# BiLSTM

root@autodl-container-7f1e4ca61f-33877ed2:~/hate/hate-speech-detection-survey# python src/training/main.py

[nltk\_data] Downloading package stopwords to /root/nltk\_data...

[nltk\_data] Package stopwords is already up-to-date!

Device: cuda

file loaded and formatted..

data split into train/val/test

field objects created

fields and dataset object created

vocabulary built..

dataloaders created..

model\_type: BiLSTM

optimizer\_type: adam

loss\_criterion: bcelosswithlogits

learning rate: 0.001

epochs: 10

patience\_es: 5

scheduler\_type:

patience\_lr: 5

save\_condition: acc

Model BiLSTM loaded on cuda

Loss used: BCEWithLogitsLoss()

Optimizer used: Adam (

Parameter Group 0

amsgrad: False

betas: (0.9, 0.999)

eps: 1e-08

lr: 0.001

weight\_decay: 0

)

Scheduler used: None

\*\*TRAINING\*\*

Epoch 0/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 43.56it/s]

train Loss: 0.6219 Acc: 0.4175

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 124.46it/s]

val Loss: 0.5804 Acc: 0.5510

Epoch complete in 2.1s

Epoch 1/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:02<00:00, 39.34it/s]

train Loss: 0.5272 Acc: 0.6476

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 121.20it/s]

val Loss: 0.5560 Acc: 0.6504

Epoch complete in 2.3s

Epoch 2/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:02<00:00, 38.64it/s]

train Loss: 0.4086 Acc: 0.7749

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 113.99it/s]

val Loss: 0.6032 Acc: 0.6701

Epoch complete in 2.4s

Epoch 3/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:02<00:00, 40.56it/s]

train Loss: 0.2869 Acc: 0.8613

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 124.65it/s]

val Loss: 0.6336 Acc: 0.6862

Epoch complete in 2.2s

Epoch 4/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:02<00:00, 40.81it/s]

train Loss: 0.1821 Acc: 0.9243

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 111.49it/s]

val Loss: 0.7772 Acc: 0.6877

Epoch complete in 2.2s

Epoch 5/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:02<00:00, 40.33it/s]

train Loss: 0.1112 Acc: 0.9545

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 112.28it/s]

val Loss: 0.9752 Acc: 0.6740

Epoch complete in 2.3s

EarlyStopping counter: 1 out of 5

Epoch 6/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:02<00:00, 40.71it/s]

train Loss: 0.0747 Acc: 0.9737

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 113.13it/s]

val Loss: 1.0914 Acc: 0.6850

Epoch complete in 2.2s

EarlyStopping counter: 2 out of 5

Epoch 7/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:02<00:00, 39.29it/s]

train Loss: 0.0533 Acc: 0.9800

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 124.13it/s]

val Loss: 1.1743 Acc: 0.6745

Epoch complete in 2.3s

EarlyStopping counter: 3 out of 5

Epoch 8/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:02<00:00, 38.14it/s]

train Loss: 0.0476 Acc: 0.9821

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 96.88it/s]

val Loss: 1.1664 Acc: 0.6826

Epoch complete in 2.4s

EarlyStopping counter: 4 out of 5

Epoch 9/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:02<00:00, 38.60it/s]

train Loss: 0.0327 Acc: 0.9867

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 108.82it/s]

val Loss: 1.3599 Acc: 0.6701

Epoch complete in 2.4s

EarlyStopping counter: 5 out of 5

Early stopping

Training complete in 0m 23s

Best val Acc: 0.687700

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████████████████████████████████████████████████████████████████| 860/860 [00:01<00:00, 472.51it/s]

Test stats - Loss: 0.6279 Acc: 70.14%

Inference on Testset complete in 1.8s

Model saved at saved-models/BiLSTM\_2025-04-26\_21-45-55\_trained\_testAcc=0.7014.pth

Training graph saved at figures/BiLSTM\_losses&acc\_2025-04-26\_21-45-55\_testAcc=0.7014.png

Confusion Matrix saved at figures/BiLSTM\_CM\_2025-04-26\_21-45-55\_testAcc=0.7014.png

Accuracy: 77.674%

Macro F1-score: 70.14%

Binary F1-score: 55.14%

MSE: 0.223

precision recall f1-score support

class 0 0.82 0.89 0.85 620

class 1 0.63 0.49 0.55 240

accuracy 0.78 860

macro avg 0.72 0.69 0.70 860

weighted avg 0.77 0.78 0.77 860说明了什么

这段输出展示了一个基于双向长短期记忆网络（BiLSTM）的仇恨言论检测模型的训练和测试过程，下面从不同阶段详细分析其结果所代表的含义：

### 1. 训练前的参数设置

```plaintext

model\_type: BiLSTM

optimizer\_type: adam

loss\_criterion: bcelosswithlogits

learning rate: 0.001

epochs: 10

patience\_es: 5

scheduler\_type:

patience\_lr: 5

save\_condition: acc

```

- \*\*模型类型（`model\_type`）\*\*：使用了双向长短期记忆网络（BiLSTM），这种模型能够捕捉序列数据中的前后文信息，适合处理自然语言文本。

- \*\*优化器（`optimizer\_type`）\*\*：采用了 Adam 优化器，它结合了 AdaGrad 和 RMSProp 的优点，在训练过程中自适应地调整每个参数的学习率。

- \*\*损失函数（`loss\_criterion`）\*\*：使用了带 logits 的二元交叉熵损失函数（`BCEWithLogitsLoss`），适用于二分类问题。

- \*\*学习率（`learning rate`）\*\*：设置为 0.001，控制模型参数更新的步长。

- \*\*训练轮数（`epochs`）\*\*：计划训练 10 个 epoch，即对整个训练数据集进行 10 次迭代。

- \*\*早停耐心值（`patience\_es`）\*\*：设置为 5，如果验证集的准确率在连续 5 个 epoch 内没有提升，则提前停止训练，防止过拟合。

- \*\*学习率调度器（`scheduler\_type`）\*\*：未使用学习率调度器，意味着学习率在训练过程中保持不变。

- \*\*保存条件（`save\_condition`）\*\*：根据验证集的准确率（`acc`）来保存最佳模型。

### 2. 训练阶段

```plaintext

Epoch 0/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 43.56it/s]

train Loss: 0.6219 Acc: 0.4175

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 124.46it/s]

val Loss: 0.5804 Acc: 0.5510

Epoch complete in 2.1s

```

- \*\*训练集损失（`train Loss`）\*\*：反映了模型在训练集上的预测误差，随着训练的进行，损失值逐渐下降，说明模型在不断学习。

- \*\*训练集准确率（`train Acc`）\*\*：表示模型在训练集上的分类准确率，随着训练的进行，准确率逐渐提高。

- \*\*验证集损失（`val Loss`）\*\*：在第 4 个 epoch 之后开始上升，这可能是过拟合的迹象，即模型在训练集上表现越来越好，但在验证集上的泛化能力下降。

- \*\*验证集准确率（`val Acc`）\*\*：在第 4 个 epoch 达到最高值 0.6877，之后开始波动或下降，结合早停机制，模型在验证集准确率不再提升时提前停止训练。

- \*\*早停机制（`EarlyStopping`）\*\*：当验证集准确率连续 5 个 epoch 没有提升时，训练提前停止，避免了过拟合的进一步恶化。

### 3. 测试阶段

```plaintext

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████████████████████████████████████████████████████████████████| 860/860 [00:01<00:00, 472.51it/s]

Test stats - Loss: 0.6279 Acc: 70.14%

Inference on Testset complete in 1.8s

```

- \*\*测试集损失（`Test stats - Loss`）\*\*：为 0.6279，衡量了模型在测试集上的预测误差。

- \*\*测试集准确率（`Test stats - Acc`）\*\*：达到 70.14%，表示模型在测试集上的分类准确率。

### 4. 模型评估指标

```plaintext

Accuracy: 77.674%

Macro F1-score: 70.14%

Binary F1-score: 55.14%

MSE: 0.223

precision recall f1-score support

class 0 0.82 0.89 0.85 620

class 1 0.63 0.49 0.55 240

accuracy 0.78 860

macro avg 0.72 0.69 0.70 860

weighted avg 0.77 0.78 0.77 860

```

- \*\*准确率（`Accuracy`）\*\*：整体分类准确率为 77.674%，但需要注意数据集是否存在类别不平衡问题，因为准确率可能会受到主导类别的影响。

- \*\*宏 F1 分数（`Macro F1-score`）\*\*：为 70.14%，对所有类别平等对待，综合考虑了精确率和召回率。

- \*\*二元 F1 分数（`Binary F1-score`）\*\*：为 55.14%，可能用于特定的二分类评估。

- \*\*均方误差（`MSE`）\*\*：为 0.223，衡量了预测值与真实值之间的平均误差。

- \*\*精确率（`precision`）、召回率（`recall`）和 F1 分数（`f1-score`）\*\*：按类别展示，类别 0 的各项指标都较高，而类别 1 的召回率较低，说明模型在识别类别 1 时存在一定困难，可能是由于数据集类别不平衡导致的。

### 5. 模型保存

```plaintext

Model saved at saved-models/BiLSTM\_2025-04-26\_21-45-55\_trained\_testAcc=0.7014.pth

Training graph saved at figures/BiLSTM\_losses&acc\_2025-04-26\_21-45-55\_testAcc=0.7014.png

Confusion Matrix saved at figures/BiLSTM\_CM\_2025-04-26\_21-45-55\_testAcc=0.7014.png

```

- 模型以 `.pth` 格式保存，方便后续使用。

- 训练过程中的损失和准确率曲线以图片形式保存，有助于可视化模型的训练过程。

- 混淆矩阵也以图片形式保存，用于直观展示模型在各个类别上的分类情况。

### 总结

- 模型在训练集上的表现不断提升，但在验证集上出现了过拟合的迹象，早停机制有效地避免了过拟合的进一步恶化。

- 测试集上的准确率为 70.14%，但类别 1 的召回率较低，可能需要处理数据集的类别不平衡问题，例如采用过采样、欠采样或使用加权损失函数等方法。

- 保存的模型、训练曲线和混淆矩阵可以用于后续的分析和优化。

# PyramidCNN

root@autodl-container-7f1e4ca61f-33877ed2:~/hate/hate-speech-detection-survey# python src/training/main.py

[nltk\_data] Downloading package stopwords to /root/nltk\_data...

[nltk\_data] Package stopwords is already up-to-date!

Device: cuda

file loaded and formatted..

data split into train/val/test

field objects created

fields and dataset object created

vocabulary built..

dataloaders created..

model\_type: PyramidCNN

optimizer\_type: adam

loss\_criterion: bcelosswithlogits

learning rate: 0.001

epochs: 10

patience\_es: 5

scheduler\_type:

patience\_lr: 5

save\_condition: acc

Context size is 2

Model PyramidCNN loaded on cuda

Loss used: BCEWithLogitsLoss()

Optimizer used: Adam (

Parameter Group 0

amsgrad: False

betas: (0.9, 0.999)

eps: 1e-08

lr: 0.001

weight\_decay: 0

)

Scheduler used: None

\*\*TRAINING\*\*

Epoch 0/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 63.18it/s]

train Loss: 0.6421 Acc: 0.4015

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 195.44it/s]

val Loss: 0.6436 Acc: 0.3956

Epoch complete in 1.4s

Epoch 1/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 71.70it/s]

train Loss: 0.6302 Acc: 0.4015

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 225.58it/s]

val Loss: 0.6409 Acc: 0.3956

Epoch complete in 1.3s

Epoch 2/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 74.75it/s]

train Loss: 0.6267 Acc: 0.4015

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 197.48it/s]

val Loss: 0.6420 Acc: 0.3956

Epoch complete in 1.2s

Epoch 3/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 77.10it/s]

train Loss: 0.6237 Acc: 0.4015

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 226.49it/s]

val Loss: 0.6322 Acc: 0.3956

Epoch complete in 1.2s

Epoch 4/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 77.13it/s]

train Loss: 0.6153 Acc: 0.4035

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 219.30it/s]

val Loss: 0.6315 Acc: 0.3956

Epoch complete in 1.2s

Epoch 5/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 74.84it/s]

train Loss: 0.5940 Acc: 0.4054

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 183.34it/s]

val Loss: 0.6218 Acc: 0.3956

Epoch complete in 1.2s

Epoch 6/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 70.69it/s]

train Loss: 0.5613 Acc: 0.4391

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 177.95it/s]

val Loss: 0.6269 Acc: 0.5651

Epoch complete in 1.3s

Epoch 7/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 74.06it/s]

train Loss: 0.5151 Acc: 0.6293

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 188.30it/s]

val Loss: 0.5863 Acc: 0.5546

Epoch complete in 1.3s

EarlyStopping counter: 1 out of 5

Epoch 8/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 77.01it/s]

train Loss: 0.4626 Acc: 0.7072

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 182.33it/s]

val Loss: 0.5983 Acc: 0.6586

Epoch complete in 1.2s

Epoch 9/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 72.25it/s]

train Loss: 0.4297 Acc: 0.7513

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 192.42it/s]

val Loss: 0.5754 Acc: 0.6663

Epoch complete in 1.3s

Training complete in 0m 13s

Best val Acc: 0.666300

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████████████████████████████████████████████████████████████████| 860/860 [00:01<00:00, 553.65it/s]

Test stats - Loss: 0.4971 Acc: 68.82%

Inference on Testset complete in 1.6s

Model saved at saved-models/PyramidCNN\_2025-04-26\_21-54-44\_trained\_testAcc=0.6882.pth

Training graph saved at figures/PyramidCNN\_losses&acc\_2025-04-26\_21-54-44\_testAcc=0.6882.png

Confusion Matrix saved at figures/PyramidCNN\_CM\_2025-04-26\_21-54-44\_testAcc=0.6882.png

Accuracy: 78.372%

Macro F1-score: 68.82%

Binary F1-score: 51.562%

MSE: 0.216

precision recall f1-score support

class 0 0.80 0.93 0.86 620

class 1 0.69 0.41 0.52 240

accuracy 0.78 860

macro avg 0.75 0.67 0.69 860

weighted avg 0.77 0.78 0.76 860

root@autodl-container-7f1e4ca61f-33877ed2:~/hate/hate-speech-detection-survey# 说明了什么

这段输出展示了使用 `PyramidCNN` 模型进行仇恨言论检测任务的训练和测试过程，下面从不同阶段详细分析其结果所代表的含义：

### 1. 训练前的参数设置

```plaintext

model\_type: PyramidCNN

optimizer\_type: adam

loss\_criterion: bcelosswithlogits

learning rate: 0.001

epochs: 10

patience\_es: 5

scheduler\_type:

patience\_lr: 5

save\_condition: acc

```

- \*\*模型类型（`model\_type`）\*\*：采用 `PyramidCNN` 模型，这是一种卷积神经网络架构，可能利用了金字塔结构来处理不同尺度的特征。

- \*\*优化器（`optimizer\_type`）\*\*：使用 Adam 优化器，它自适应地调整每个参数的学习率，有助于更高效地训练模型。

- \*\*损失函数（`loss\_criterion`）\*\*：选用带 logits 的二元交叉熵损失函数（`BCEWithLogitsLoss`），适用于二分类问题。

- \*\*学习率（`learning rate`）\*\*：设置为 0.001，控制模型参数更新的步长。

- \*\*训练轮数（`epochs`）\*\*：计划进行 10 个 epoch 的训练，即对整个训练数据集迭代 10 次。

- \*\*早停耐心值（`patience\_es`）\*\*：设为 5，如果验证集的准确率在连续 5 个 epoch 内没有提升，训练将提前停止，以防止过拟合。

- \*\*学习率调度器（`scheduler\_type`）\*\*：未使用学习率调度器，意味着学习率在训练过程中保持不变。

- \*\*保存条件（`save\_condition`）\*\*：根据验证集的准确率（`acc`）来保存最佳模型。

### 2. 训练阶段

```plaintext

Epoch 0/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 63.18it/s]

train Loss: 0.6421 Acc: 0.4015

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 195.44it/s]

val Loss: 0.6436 Acc: 0.3956

Epoch complete in 1.4s

```

- \*\*训练初期\*\*：在前几个 epoch 中，训练集损失和验证集损失都较高，准确率较低，说明模型还处于学习阶段，尚未很好地拟合数据。例如，在 epoch 0 时，训练集损失为 0.6421，准确率为 0.4015；验证集损失为 0.6436，准确率为 0.3956。

- \*\*学习过程\*\*：随着训练的进行，训练集损失逐渐下降，准确率逐渐提高。从 epoch 6 开始，模型在训练集上的准确率有了明显提升，例如在 epoch 7 时，训练集损失降至 0.5151，准确率达到 0.6293。

- \*\*验证集表现\*\*：验证集的准确率在前期波动较大，直到 epoch 6 才开始有较为明显的提升。不过，在训练后期，验证集损失和准确率的提升速度不如训练集，这可能是过拟合的早期迹象。

### 3. 早停机制

```plaintext

EarlyStopping counter: 1 out of 5

```

早停机制在训练过程中发挥了作用，当验证集准确率在连续多个 epoch 内没有提升时，计数器开始计数。虽然在这个训练过程中没有因为早停而提前结束训练，但早停机制有助于避免模型过度拟合训练数据，提高模型的泛化能力。

### 4. 测试阶段

```plaintext

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████████████████████████████████████████████████████████████████| 860/860 [00:01<00:00, 553.65it/s]

Test stats - Loss: 0.4971 Acc: 68.82%

Inference on Testset complete in 1.6s

```

- \*\*测试集损失（`Test stats - Loss`）\*\*：为 0.4971，相较于训练初期的损失有了显著下降，说明模型在测试集上的预测误差有所减小。

- \*\*测试集准确率（`Test stats - Acc`）\*\*：达到 68.82%，表示模型在测试集上能够正确分类约 68.82% 的样本。

### 5. 模型评估指标

```plaintext

Accuracy: 78.372%

Macro F1-score: 68.82%

Binary F1-score: 51.562%

MSE: 0.216

precision recall f1-score support

class 0 0.80 0.93 0.86 620

class 1 0.69 0.41 0.52 240

accuracy 0.78 860

macro avg 0.75 0.67 0.69 860

weighted avg 0.77 0.78 0.76 860

```

- \*\*准确率（`Accuracy`）\*\*：整体分类准确率为 78.372%，但需要注意数据集是否存在类别不平衡问题，因为准确率可能会受到主导类别的影响。

- \*\*宏 F1 分数（`Macro F1-score`）\*\*：为 68.82%，对所有类别平等对待，综合考虑了精确率和召回率。

- \*\*二元 F1 分数（`Binary F1-score`）\*\*：为 51.562%，可能用于特定的二分类评估。

- \*\*均方误差（`MSE`）\*\*：为 0.216，衡量了预测值与真实值之间的平均误差。

- \*\*精确率（`precision`）、召回率（`recall`）和 F1 分数（`f1-score`）\*\*：按类别展示，类别 0 的各项指标都较高，而类别 1 的召回率较低，说明模型在识别类别 1 时存在一定困难，可能是由于数据集类别不平衡导致的。

### 6. 模型保存

```plaintext

Model saved at saved-models/PyramidCNN\_2025-04-26\_21-54-44\_trained\_testAcc=0.6882.pth

Training graph saved at figures/PyramidCNN\_losses&acc\_2025-04-26\_21-54-44\_testAcc=0.6882.png

Confusion Matrix saved at figures/PyramidCNN\_CM\_2025-04-26\_21-54-44\_testAcc=0.6882.png

```

- 模型以 `.pth` 格式保存，方便后续使用。

- 训练过程中的损失和准确率曲线以图片形式保存，有助于可视化模型的训练过程。

- 混淆矩阵也以图片形式保存，用于直观展示模型在各个类别上的分类情况。

### 总结

- 模型在训练集上的表现随着训练的进行逐渐提升，但在验证集上的提升相对较慢，可能存在一定的过拟合风险。

- 早停机制有助于防止过拟合，但可以考虑进一步优化模型结构或调整超参数，以提高模型的泛化能力。

- 测试集上的准确率为 68.82%，但类别 1 的召回率较低，需要关注数据集的类别不平衡问题，可以尝试采用过采样、欠采样或使用加权损失函数等方法来解决。

- 保存的模型、训练曲线和混淆矩阵可以用于后续的分析和优化。

# HybridCNNLSTM

root@autodl-container-7f1e4ca61f-33877ed2:~/hate/hate-speech-detection-survey# python src/training/main.py

[nltk\_data] Downloading package stopwords to /root/nltk\_data...

[nltk\_data] Package stopwords is already up-to-date!

Device: cuda

file loaded and formatted..

data split into train/val/test

field objects created

fields and dataset object created

vocabulary built..

dataloaders created..

model\_type: HybridCNNLSTM

optimizer\_type: adam

loss\_criterion: bcelosswithlogits

learning rate: 0.001

epochs: 10

patience\_es: 5

scheduler\_type:

patience\_lr: 5

save\_condition: acc

Model HybridCNNLSTM loaded on cuda

Loss used: BCEWithLogitsLoss()

Optimizer used: Adam (

Parameter Group 0

amsgrad: False

betas: (0.9, 0.999)

eps: 1e-08

lr: 0.001

weight\_decay: 0

)

Scheduler used: None

\*\*TRAINING\*\*

Epoch 0/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 47.22it/s]

train Loss: 0.6298 Acc: 0.4015

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 140.45it/s]

val Loss: 0.6341 Acc: 0.3956

Epoch complete in 1.9s

Epoch 1/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 51.62it/s]

train Loss: 0.5702 Acc: 0.5495

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 143.21it/s]

val Loss: 0.5959 Acc: 0.5425

Epoch complete in 1.8s

Epoch 2/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 49.69it/s]

train Loss: 0.4866 Acc: 0.6889

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 143.07it/s]

val Loss: 0.6233 Acc: 0.6649

Epoch complete in 1.8s

Epoch 3/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 45.33it/s]

train Loss: 0.3864 Acc: 0.7929

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 111.64it/s]

val Loss: 0.6059 Acc: 0.6552

Epoch complete in 2.0s

EarlyStopping counter: 1 out of 5

Epoch 4/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 43.15it/s]

train Loss: 0.2901 Acc: 0.8597

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 151.81it/s]

val Loss: 0.6926 Acc: 0.6635

Epoch complete in 2.1s

EarlyStopping counter: 2 out of 5

Epoch 5/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 44.77it/s]

train Loss: 0.2081 Acc: 0.9091

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 127.38it/s]

val Loss: 0.8090 Acc: 0.6600

Epoch complete in 2.0s

EarlyStopping counter: 3 out of 5

Epoch 6/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 46.23it/s]

train Loss: 0.1534 Acc: 0.9346

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 108.11it/s]

val Loss: 1.0414 Acc: 0.6395

Epoch complete in 2.0s

EarlyStopping counter: 4 out of 5

Epoch 7/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 43.85it/s]

train Loss: 0.1126 Acc: 0.9548

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 116.62it/s]

val Loss: 1.1610 Acc: 0.6593

Epoch complete in 2.1s

EarlyStopping counter: 5 out of 5

Early stopping

Training complete in 0m 16s

Best val Acc: 0.664900

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████████████████████████████████████████████████████████████████| 860/860 [00:02<00:00, 401.46it/s]

Test stats - Loss: 0.5633 Acc: 66.51%

Inference on Testset complete in 2.1s

Model saved at saved-models/HybridCNNLSTM\_2025-04-26\_22-00-23\_trained\_testAcc=0.6651.pth

Training graph saved at figures/HybridCNNLSTM\_losses&acc\_2025-04-26\_22-00-23\_testAcc=0.6651.png

Confusion Matrix saved at figures/HybridCNNLSTM\_CM\_2025-04-26\_22-00-23\_testAcc=0.6651.png

Accuracy: 74.419%

Macro F1-score: 66.506%

Binary F1-score: 50.226%

MSE: 0.256

precision recall f1-score support

class 0 0.80 0.85 0.83 620

class 1 0.55 0.46 0.50 240

accuracy 0.74 860

macro avg 0.68 0.66 0.67 860

weighted avg 0.73 0.74 0.74 860

root@autodl-container-7f1e4ca61f-33877ed2:~/hate/hate-speech-detection-survey# 这说明了什么

这段输出展示了使用 `HybridCNNLSTM` 模型进行仇恨言论检测任务的训练和测试过程，下面从不同方面详细分析其结果的含义：

### 1. 训练前的配置信息

```plaintext

model\_type: HybridCNNLSTM

optimizer\_type: adam

loss\_criterion: bcelosswithlogits

learning rate: 0.001

epochs: 10

patience\_es: 5

scheduler\_type:

patience\_lr: 5

save\_condition: acc

```

- \*\*模型类型\*\*：采用了 `HybridCNNLSTM` 模型，结合了卷积神经网络（CNN）和长短期记忆网络（LSTM）的特点，CNN 可用于提取局部特征，LSTM 能处理序列数据中的长期依赖关系。

- \*\*优化器\*\*：使用 Adam 优化器，它结合了 AdaGrad 和 RMSProp 的优点，自适应地调整每个参数的学习率，有助于模型更高效地收敛。

- \*\*损失函数\*\*：选用 `BCEWithLogitsLoss`，适用于二分类问题，能更好地处理模型输出的 logits 并计算损失。

- \*\*学习率\*\*：设置为 0.001，这是一个常见的学习率初始值，控制着模型参数更新的步长。

- \*\*训练轮数\*\*：计划进行 10 个 epoch 的训练，即对整个训练数据集迭代 10 次。

- \*\*早停机制\*\*：`patience\_es` 设置为 5，意味着如果验证集的准确率在连续 5 个 epoch 内没有提升，训练将提前停止，防止过拟合。

- \*\*学习率调度器\*\*：未使用学习率调度器，学习率在训练过程中保持不变。

- \*\*保存条件\*\*：根据验证集的准确率来保存最佳模型。

### 2. 训练阶段

```plaintext

Epoch 0/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 47.22it/s]

train Loss: 0.6298 Acc: 0.4015

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 140.45it/s]

val Loss: 0.6341 Acc: 0.3956

Epoch complete in 1.9s

```

- \*\*训练初期\*\*：在训练开始时，训练集和验证集的损失都较高，准确率较低，说明模型还处于学习的初始阶段，尚未很好地拟合数据。例如，在 epoch 0 时，训练集损失为 0.6298，准确率为 0.4015；验证集损失为 0.6341，准确率为 0.3956。

- \*\*学习过程\*\*：随着训练的进行，训练集的损失迅速下降，准确率显著提高。如在 epoch 3 时，训练集损失降至 0.3864，准确率达到 0.7929，表明模型在训练集上的学习效果良好。

- \*\*过拟合迹象\*\*：从 epoch 3 开始，验证集的损失不再下降，反而逐渐上升，尽管准确率有一定波动，但整体提升不明显，这是过拟合的典型表现。模型在训练集上的表现越来越好，但在验证集上的泛化能力逐渐变差。

- \*\*早停机制触发\*\*：由于验证集准确率在连续 5 个 epoch 内没有提升，早停机制在 epoch 7 触发，训练提前结束，避免了模型进一步过拟合。

### 3. 测试阶段

```plaintext

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████████████████████████████████████████████████████████████████| 860/860 [00:02<00:00, 401.46it/s]

Test stats - Loss: 0.5633 Acc: 66.51%

Inference on Testset complete in 2.1s

```

- \*\*测试集损失和准确率\*\*：测试集损失为 0.5633，准确率为 66.51%。测试集的损失和准确率可以反映模型在未见过的数据上的泛化能力。由于训练过程中出现了过拟合，测试集的准确率相对训练集的最终准确率有所下降。

### 4. 模型评估指标

```plaintext

Accuracy: 74.419%

Macro F1-score: 66.506%

Binary F1-score: 50.226%

MSE: 0.256

precision recall f1-score support

class 0 0.80 0.85 0.83 620

class 1 0.55 0.46 0.50 240

accuracy 0.74 860

macro avg 0.68 0.66 0.67 860

weighted avg 0.73 0.74 0.74 860

```

- \*\*准确率（Accuracy）\*\*：整体准确率为 74.419%，但由于数据集可能存在类别不平衡问题（类别 0 有 620 个样本，类别 1 有 240 个样本），准确率可能会受到主导类别的影响，不能完全反映模型的性能。

- \*\*F1 分数\*\*：

- \*\*宏 F1 分数（Macro F1-score）\*\*：为 66.506%，对所有类别平等对待，综合考虑了精确率和召回率。

- \*\*二元 F1 分数（Binary F1-score）\*\*：为 50.226%，可能用于特定的二分类评估。

- 从类别角度看，类别 0 的精确率、召回率和 F1 分数都较高，而类别 1 的各项指标相对较低，说明模型在识别类别 1 时存在困难，可能是由于类别不平衡导致模型对少数类别的学习不足。

- \*\*均方误差（MSE）\*\*：为 0.256，衡量了模型预测值与真实值之间的平均误差。

### 5. 模型保存

```plaintext

Model saved at saved-models/HybridCNNLSTM\_2025-04-26\_22-00-23\_trained\_testAcc=0.6651.pth

Training graph saved at figures/HybridCNNLSTM\_losses&acc\_2025-04-26\_22-00-23\_testAcc=0.6651.png

Confusion Matrix saved at figures/HybridCNNLSTM\_CM\_2025-04-26\_22-00-23\_testAcc=0.6651.png

```

- 模型以 `.pth` 格式保存，方便后续使用和进一步优化。

- 训练过程中的损失和准确率曲线以图片形式保存，有助于可视化模型的训练过程，分析模型的学习情况。

- 混淆矩阵也以图片形式保存，可直观展示模型在各个类别上的分类情况，帮助发现模型的优势和不足。

### 总结

- \*\*模型表现\*\*：`HybridCNNLSTM` 模型在训练集上的学习效果较好，但出现了过拟合现象，导致在验证集和测试集上的泛化能力受限。

- \*\*类别不平衡问题\*\*：数据集的类别不平衡影响了模型对少数类别的识别能力，类别 1 的性能明显低于类别 0。

- \*\*改进方向\*\*：为提高模型性能，可以考虑以下措施：

- 调整模型结构，如增加正则化项（如 Dropout）、减少模型复杂度，以缓解过拟合问题。

- 处理类别不平衡问题，如采用过采样、欠采样或使用加权损失函数等方法。

- 调整超参数，如学习率、训练轮数等，以找到更合适的模型配置。

# HybridLSTMCNN

root@autodl-container-09dd4b98a3-2aa539b6:~/hate/hate-speech-detection-survey# python src/training/main.py --fix\_length 20

[nltk\_data] Downloading package stopwords to /root/nltk\_data...

[nltk\_data] Package stopwords is already up-to-date!

Device: cuda

file loaded and formatted..

data split into train/val/test

field objects created

fields and dataset object created

vocabulary built..

dataloaders created..

model\_type: HybridLSTMCNN

optimizer\_type: adam

loss\_criterion: bcelosswithlogits

learning rate: 0.001

epochs: 10

patience\_es: 5

scheduler\_type:

patience\_lr: 5

save\_condition: acc

Model HybridLSTMCNN loaded on cuda

Loss used: BCEWithLogitsLoss()

Optimizer used: Adam (

Parameter Group 0

amsgrad: False

betas: (0.9, 0.999)

eps: 1e-08

lr: 0.001

weight\_decay: 0

)

Scheduler used: None

\*\*TRAINING\*\*

Epoch 0/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 61.44it/s]

train Loss: 0.6405 Acc: 0.4015

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 182.92it/s]

val Loss: 0.6481 Acc: 0.3956

Epoch complete in 1.5s

Epoch 1/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 54.33it/s]

train Loss: 0.6205 Acc: 0.4103

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 187.44it/s]

val Loss: 0.6112 Acc: 0.3991

Epoch complete in 1.7s

Epoch 2/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 55.68it/s]

train Loss: 0.5592 Acc: 0.5624

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 191.67it/s]

val Loss: 0.5644 Acc: 0.5799

Epoch complete in 1.6s

Epoch 3/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 63.49it/s]

train Loss: 0.4968 Acc: 0.6671

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 181.21it/s]

val Loss: 0.5810 Acc: 0.6474

Epoch complete in 1.4s

Epoch 4/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 67.77it/s]

train Loss: 0.4422 Acc: 0.7381

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 187.85it/s]

val Loss: 0.5994 Acc: 0.6547

Epoch complete in 1.4s

Epoch 5/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 59.31it/s]

train Loss: 0.3704 Acc: 0.8039

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 219.43it/s]

val Loss: 0.7138 Acc: 0.6811

Epoch complete in 1.5s

Epoch 6/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 48.29it/s]

train Loss: 0.2956 Acc: 0.8555

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 187.16it/s]

val Loss: 0.7108 Acc: 0.6806

Epoch complete in 1.9s

EarlyStopping counter: 1 out of 5

Epoch 7/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 49.43it/s]

train Loss: 0.2126 Acc: 0.9080

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 191.04it/s]

val Loss: 0.9962 Acc: 0.6757

Epoch complete in 1.8s

EarlyStopping counter: 2 out of 5

Epoch 8/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 55.93it/s]

train Loss: 0.1601 Acc: 0.9328

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 193.79it/s]

val Loss: 1.0985 Acc: 0.6779

Epoch complete in 1.6s

EarlyStopping counter: 3 out of 5

Epoch 9/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 50.22it/s]

train Loss: 0.1213 Acc: 0.9500

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 196.95it/s]

val Loss: 1.3105 Acc: 0.6767

Epoch complete in 1.8s

EarlyStopping counter: 4 out of 5

Training complete in 0m 16s

Best val Acc: 0.681100

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████████████████████████████████████████████████████████████████| 860/860 [00:02<00:00, 392.49it/s]

Test stats - Loss: 0.6795 Acc: 68.48%

Inference on Testset complete in 2.2s

Model saved at saved-models/HybridLSTMCNN\_2025-05-01\_17-33-07\_trained\_testAcc=0.6848.pth

Training graph saved at figures/HybridLSTMCNN\_losses&acc\_2025-05-01\_17-33-07\_testAcc=0.6848.png

Confusion Matrix saved at figures/HybridLSTMCNN\_CM\_2025-05-01\_17-33-07\_testAcc=0.6848.png

Accuracy: 74.535%

Macro F1-score: 68.477%

Binary F1-score: 54.658%

MSE: 0.255

precision recall f1-score support

class 0 0.82 0.82 0.82 620

class 1 0.54 0.55 0.55 240

accuracy 0.75 860

macro avg 0.68 0.69 0.68 860

weighted avg 0.75 0.75 0.75 860

root@autodl-container-09dd4b98a3-2aa539b6:~/hate/hate-speech-detection-survey# 说明了什么

从输出结果可以看出模型训练、测试的整个流程以及最终的性能表现，下面详细分析各项结果所代表的含义。

### 1. 训练前的配置信息

```plaintext

model\_type: HybridLSTMCNN

optimizer\_type: adam

loss\_criterion: bcelosswithlogits

learning rate: 0.001

epochs: 10

patience\_es: 5

scheduler\_type:

patience\_lr: 5

save\_condition: acc

```

- \*\*模型类型\*\*：使用的是 `HybridLSTMCNN` 模型，结合了 LSTM 和 CNN 的特性，适合处理序列数据，比如文本数据。

- \*\*优化器\*\*：采用 Adam 优化器，这是一种自适应学习率的优化算法，在很多深度学习任务中表现良好。

- \*\*损失函数\*\*：运用 `BCEWithLogitsLoss`，适用于二分类问题，会在内部进行 sigmoid 激活，然后计算二元交叉熵损失。

- \*\*学习率\*\*：设定为 0.001，这是常见的学习率初始值，用于控制模型参数更新的步长。

- \*\*训练轮数\*\*：计划进行 10 个 epoch 的训练，也就是让模型对整个训练数据集迭代 10 次。

- \*\*早停机制\*\*：`patience\_es` 设为 5，意味着如果验证集的准确率在连续 5 个 epoch 内没有提升，训练就会提前停止，以此防止过拟合。

- \*\*学习率调度器\*\*：未使用学习率调度器，所以学习率在训练过程中保持不变。

- \*\*保存条件\*\*：依据验证集的准确率来保存最佳模型。

### 2. 训练过程

```plaintext

Epoch 0/9 | Trainings remaining: 1

----------

current lr: 0.001

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:01<00:00, 61.44it/s]

train Loss: 0.6405 Acc: 0.4015

Processing batch 21: 100%|█████████████████████████████████████████████████████████████████████| 21/21 [00:00<00:00, 182.92it/s]

val Loss: 0.6481 Acc: 0.3956

Epoch complete in 1.5s

```

- \*\*训练初期\*\*：在训练开始时，训练集和验证集的损失都比较高，准确率较低，这表明模型还处于学习的初始阶段，尚未很好地拟合数据。例如，在 epoch 0 时，训练集损失为 0.6405，准确率为 0.4015；验证集损失为 0.6481，准确率为 0.3956。

- \*\*学习过程\*\*：随着训练的推进，训练集的损失逐渐下降，准确率不断提高。如在 epoch 5 时，训练集损失降至 0.3704，准确率达到 0.8039，说明模型在训练集上的学习效果良好。

- \*\*过拟合迹象\*\*：从 epoch 5 之后，验证集的损失开始上升，尽管准确率有一定波动，但整体提升不明显，这是过拟合的典型表现。也就是说，模型在训练集上的表现越来越好，但在验证集上的泛化能力逐渐变差。

- \*\*早停机制触发\*\*：由于验证集准确率在连续 4 个 epoch 内没有提升，早停机制在 epoch 9 之前虽然没有完全触发（`EarlyStopping counter: 4 out of 5`），但也显示出模型性能开始停滞。

### 3. 测试阶段

```plaintext

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████████████████████████████████████████████████████████████████| 860/860 [00:02<00:00, 392.49it/s]

Test stats - Loss: 0.6795 Acc: 68.48%

Inference on Testset complete in 2.2s

```

- \*\*测试集损失和准确率\*\*：测试集损失为 0.6795，准确率为 68.48%。测试集的损失和准确率能够反映模型在未见过的数据上的泛化能力。由于训练过程中出现了过拟合，测试集的准确率相对训练集的最终准确率有所下降。

### 4. 模型评估指标

```plaintext

Accuracy: 74.535%

Macro F1-score: 68.477%

Binary F1-score: 54.658%

MSE: 0.255

precision recall f1-score support

class 0 0.82 0.82 0.82 620

class 1 0.54 0.55 0.55 240

accuracy 0.75 860

macro avg 0.68 0.69 0.68 860

weighted avg 0.75 0.75 0.75 860

```

- \*\*准确率（Accuracy）\*\*：整体准确率为 74.535%，不过由于数据集可能存在类别不平衡问题（类别 0 有 620 个样本，类别 1 有 240 个样本），准确率可能会受到主导类别的影响，不能完全反映模型的性能。

- \*\*F1 分数\*\*：

- \*\*宏 F1 分数（Macro F1-score）\*\*：为 68.477%，对所有类别平等对待，综合考虑了精确率和召回率。

- \*\*二元 F1 分数（Binary F1-score）\*\*：为 54.658%，可能用于特定的二分类评估。

- 从类别角度看，类别 0 的精确率、召回率和 F1 分数都较高，而类别 1 的各项指标相对较低，说明模型在识别类别 1 时存在困难，可能是由于类别不平衡导致模型对少数类别的学习不足。

- \*\*均方误差（MSE）\*\*：为 0.255，衡量了模型预测值与真实值之间的平均误差。

### 5. 模型保存

```plaintext

Model saved at saved-models/HybridLSTMCNN\_2025-05-01\_17-33-07\_trained\_testAcc=0.6848.pth

Training graph saved at figures/HybridLSTMCNN\_losses&acc\_2025-05-01\_17-33-07\_testAcc=0.6848.png

Confusion Matrix saved at figures/HybridLSTMCNN\_CM\_2025-05-01\_17-33-07\_testAcc=0.6848.png

```

- 模型以 `.pth` 格式保存，方便后续使用和进一步优化。

- 训练过程中的损失和准确率曲线以图片形式保存，有助于可视化模型的训练过程，分析模型的学习情况。

- 混淆矩阵也以图片形式保存，可直观展示模型在各个类别上的分类情况，帮助发现模型的优势和不足。

### 总结

- \*\*模型表现\*\*：`HybridLSTMCNN` 模型在训练集上的学习效果较好，但出现了过拟合现象，导致在验证集和测试集上的泛化能力受限。

- \*\*类别不平衡问题\*\*：数据集的类别不平衡影响了模型对少数类别的识别能力，类别 1 的性能明显低于类别 0。

- \*\*改进方向\*\*：为提高模型性能，可以考虑以下措施：

- 调整模型结构，如增加正则化项（如 Dropout）、减少模型复杂度，以缓解过拟合问题。

- 处理类别不平衡问题，如采用过采样、欠采样或使用加权损失函数等方法。

- 调整超参数，如学习率、训练轮数等，以找到更合适的模型配置。

# DistillBert

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# python src/training/main.py --model DistillBert --loss\_criterion crossentropy --lr 2e-5

[nltk\_data] Downloading package stopwords to /root/nltk\_data...

[nltk\_data] Package stopwords is already up-to-date!

Device: cuda

file loaded and formatted..

data split into train/val/test

pad\_index 0

field objects created

fields and dataset object created

vocabulary built..

dataloaders created..

model\_type: DistillBert

optimizer\_type: adam

loss\_criterion: crossentropy

learning rate: 2e-05

epochs: 10

patience\_es: 5

scheduler\_type:

patience\_lr: 5

save\_condition: acc

Model DistillBert loaded on cuda

Loss used: CrossEntropyLoss()

Optimizer used: Adam (

Parameter Group 0

amsgrad: False

betas: (0.9, 0.999)

eps: 1e-08

lr: 2e-05

weight\_decay: 0

)

Scheduler used: None

\*\*TRAINING\*\*

Epoch 0/9 | Trainings remaining: 1

----------

current lr: 2e-05

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:16<00:00, 4.94it/s]

train Loss: 0.5236 Acc: 0.6550

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:01<00:00, 14.65it/s]

val Loss: 0.4739 Acc: 0.7292

Epoch complete in 18.3s

Epoch 1/9 | Trainings remaining: 1

----------

current lr: 2e-05

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:16<00:00, 4.94it/s]

train Loss: 0.4030 Acc: 0.7892

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:01<00:00, 13.56it/s]

val Loss: 0.4386 Acc: 0.7724

Epoch complete in 18.4s

Epoch 2/9 | Trainings remaining: 1

----------

current lr: 2e-05

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:16<00:00, 4.99it/s]

train Loss: 0.3438 Acc: 0.8334

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:01<00:00, 14.12it/s]

val Loss: 0.4611 Acc: 0.7740

Epoch complete in 18.2s

Epoch 3/9 | Trainings remaining: 1

----------

current lr: 2e-05

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:16<00:00, 5.01it/s]

train Loss: 0.2803 Acc: 0.8731

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:01<00:00, 13.64it/s]

val Loss: 0.4996 Acc: 0.7701

Epoch complete in 18.1s

EarlyStopping counter: 1 out of 5

Epoch 4/9 | Trainings remaining: 1

----------

current lr: 2e-05

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:16<00:00, 4.98it/s]

train Loss: 0.2096 Acc: 0.9158

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:01<00:00, 14.40it/s]

val Loss: 0.5683 Acc: 0.7510

Epoch complete in 18.1s

EarlyStopping counter: 2 out of 5

Epoch 5/9 | Trainings remaining: 1

----------

current lr: 2e-05

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:16<00:00, 5.04it/s]

train Loss: 0.1514 Acc: 0.9441

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:01<00:00, 13.69it/s]

val Loss: 0.6731 Acc: 0.7512

Epoch complete in 18.0s

EarlyStopping counter: 3 out of 5

Epoch 6/9 | Trainings remaining: 1

----------

current lr: 2e-05

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:16<00:00, 5.05it/s]

train Loss: 0.1026 Acc: 0.9661

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:01<00:00, 14.30it/s]

val Loss: 0.7066 Acc: 0.7592

Epoch complete in 17.9s

EarlyStopping counter: 4 out of 5

Epoch 7/9 | Trainings remaining: 1

----------

current lr: 2e-05

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:16<00:00, 4.95it/s]

train Loss: 0.0777 Acc: 0.9769

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:01<00:00, 13.24it/s]

val Loss: 0.7763 Acc: 0.7520

Epoch complete in 18.4s

EarlyStopping counter: 5 out of 5

Early stopping

Training complete in 2m 27s

Best val Acc: 0.774000

\*\*TESTING\*\*

Loss used: CrossEntropyLoss()

Processing batch 860: 100%|██████████████████████████████████████████████████████████████████| 860/860 [00:04<00:00, 177.51it/s]

Test stats - Loss: 0.3749 Acc: 80.27%

Inference on Testset complete in 4.8s

Model saved at saved-models/DistillBert\_2025-05-07\_16-28-50\_trained\_testAcc=0.8027.pth

Training graph saved at figures/DistillBert\_losses&acc\_2025-05-07\_16-28-50\_testAcc=0.8027.png

Confusion Matrix saved at figures/DistillBert\_CM\_2025-05-07\_16-28-50\_testAcc=0.8027.png

Accuracy: 85.0%

Macro F1-score: 80.272%

Binary F1-score: 70.615%

MSE: 0.15

precision recall f1-score support

class 0 0.87 0.93 0.90 620

class 1 0.78 0.65 0.71 240

accuracy 0.85 860

macro avg 0.83 0.79 0.80 860

weighted avg 0.85 0.85 0.85 860，说明了啥

从你提供的训练和测试结果来看，模型在仇恨言论检测任务上取得了较为合理的表现，以下是具体分析：

### 一、训练阶段关键发现

1. \*\*损失与准确率趋势\*\*：

- \*\*训练损失\*\*从 `0.5236` 快速下降至 `0.0777`，\*\*训练准确率\*\*从 `65.50%` 提升至 `97.69%`，显示模型对训练数据拟合良好。

- \*\*验证损失\*\*在 `Epoch 2` 达到最低 `0.4611`，验证准确率在 `Epoch 2` 达到最高 `77.40%`，随后因验证集表现未持续提升，早停机制（`patience=5`）在 `Epoch 7` 触发，避免了过拟合。

2. \*\*学习率影响\*\*：

- 使用较低的学习率 `2e-5` 后，模型收敛稳定，避免了早期因学习率过高导致的梯度爆炸或震荡（参考之前 `BCEWithLogitsLoss` 的失败案例）。

### 二、测试阶段核心指标

1. \*\*整体性能\*\*：

- \*\*测试准确率\*\* `85.0%`：模型在测试集上的整体分类准确率较高，说明泛化能力较好。

- \*\*Macro F1-score\*\* `80.27%`：平衡了正负类的平均表现，优于单一准确率指标。

- \*\*Binary F1-score\*\* `70.61%`：针对二分类任务的综合指标，反映模型对正负类的整体区分能力。

2. \*\*类别不平衡处理\*\*：

- \*\*类别0（非仇恨言论）\*\*：

- precision `0.87`（预测为非仇恨的样本中，实际为非仇恨的比例）

- recall `0.93`（实际非仇恨样本中，被正确识别的比例）

- 表现优异，说明模型对多数类的识别能力强。

- \*\*类别1（仇恨言论）\*\*：

- precision `0.78`（预测为仇恨的样本中，实际为仇恨的比例）

- recall `0.65`（实际仇恨样本中，被正确识别的比例）

- 召回率偏低，表明模型对少数类的漏判较多（部分仇恨言论未被检测出），可能受类别不平衡影响（测试集正负样本比约为 2.5:1）。

3. \*\*损失函数选择\*\*：

- 使用 `CrossEntropyLoss` 替代 `BCEWithLogitsLoss` 后，模型输出二维 logits（符合多分类/二分类规范），避免了维度不匹配错误，且训练流程正常收敛。

### 三、模型与数据适配性

1. \*\*DistillBert 模型效果\*\*：

- 轻量级预训练模型（DistillBert）在微调后能够有效提取文本特征，证明预训练模型在小数据集上的有效性。

- 平均池化（`last\_hidden\_state.mean(dim=0)`）作为特征聚合方式，简单有效，但可能未充分利用序列建模能力（如CLS token）。

2. \*\*数据预处理\*\*：

- 文本清洗（URL替换、表情符号处理）和分词逻辑正确，保证了输入数据质量。

- 固定序列长度（`fix\_length`）和填充掩码（`pad\_index=0`）处理正确，避免了无效Token的干扰。

### 四、潜在问题与改进方向

1. \*\*类别不平衡优化\*\*：

- 少数类召回率低（65%），可通过以下方式改进：

- \*\*过采样\*\*：对少数类样本进行SMOTE过采样，平衡训练集类别分布。

- \*\*损失函数加权\*\*：在 `CrossEntropyLoss` 中添加类别权重（如 `weight=torch.tensor([1, 3])` 提升少数类重要性）。

- \*\*焦点损失（Focal Loss）\*\*：降低多数类样本的训练权重，聚焦少数类。

2. \*\*模型结构调整\*\*：

- 尝试使用 `CLS token` 作为特征聚合（替代平均池化），增强关键特征提取。

- 增加 dropout 或权重衰减，缓解过拟合（当前训练集准确率过高，可能存在轻微过拟合）。

3. \*\*超参数调优\*\*：

- 引入学习率调度器（如 `ReduceLROnPlateau`），在验证集停滞时动态降低学习率。

- 调整批次大小（当前 `batch\_size=128`），可能需要更小批次以稳定梯度。

### 五、总结

当前模型在仇恨言论检测任务上达到 \*\*85% 的整体准确率\*\*，但对少数类（仇恨言论）的检测能力仍有提升空间。通过调整损失函数、处理类别不平衡和优化模型结构，可进一步提高对仇恨言论的召回率，确保模型在实际场景中的可靠性。后续可重点关注少数类样本的学习，结合领域知识优化数据预处理或尝试更复杂的模型架构（如全连接层增加隐藏层）。

# DistillBertEmotion

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# python src/training/main.py --model DistillBertEmotion --loss\_criterion crossentropy --lr 2e-5

[nltk\_data] Downloading package stopwords to /root/nltk\_data...

[nltk\_data] Package stopwords is already up-to-date!

Device: cuda

file loaded and formatted..

data split into train/val/test

pad\_index 0

field objects created

fields and dataset object created

vocabulary built..

dataloaders created..

model\_type: DistillBertEmotion

optimizer\_type: adam

loss\_criterion: crossentropy

learning rate: 2e-05

epochs: 10

patience\_es: 5

scheduler\_type:

patience\_lr: 5

save\_condition: acc

Model DistillBertEmotion loaded on cuda

Loss used: CrossEntropyLoss()

Optimizer used: Adam (

Parameter Group 0

amsgrad: False

betas: (0.9, 0.999)

eps: 1e-08

lr: 2e-05

weight\_decay: 0

)

Scheduler used: None

\*\*TRAINING\*\*

Epoch 0/9 | Trainings remaining: 1

----------

current lr: 2e-05

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:16<00:00, 4.99it/s]

train Loss: 0.5062 Acc: 0.6964

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:01<00:00, 13.45it/s]

val Loss: 0.4433 Acc: 0.7698

Epoch complete in 18.2s

Epoch 1/9 | Trainings remaining: 1

----------

current lr: 2e-05

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:16<00:00, 4.96it/s]

train Loss: 0.4034 Acc: 0.7938

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:01<00:00, 13.07it/s]

val Loss: 0.4374 Acc: 0.7651

Epoch complete in 18.4s

EarlyStopping counter: 1 out of 5

Epoch 2/9 | Trainings remaining: 1

----------

current lr: 2e-05

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:16<00:00, 4.96it/s]

train Loss: 0.3547 Acc: 0.8257

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:01<00:00, 14.29it/s]

val Loss: 0.4564 Acc: 0.7675

Epoch complete in 18.2s

EarlyStopping counter: 2 out of 5

Epoch 3/9 | Trainings remaining: 1

----------

current lr: 2e-05

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:16<00:00, 5.01it/s]

train Loss: 0.2958 Acc: 0.8651

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:01<00:00, 13.57it/s]

val Loss: 0.4779 Acc: 0.7671

Epoch complete in 18.1s

EarlyStopping counter: 3 out of 5

Epoch 4/9 | Trainings remaining: 1

----------

current lr: 2e-05

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:17<00:00, 4.88it/s]

train Loss: 0.2260 Acc: 0.9064

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:01<00:00, 14.48it/s]

val Loss: 0.5654 Acc: 0.7526

Epoch complete in 18.5s

EarlyStopping counter: 4 out of 5

Epoch 5/9 | Trainings remaining: 1

----------

current lr: 2e-05

Processing batch 83: 100%|██████████████████████████████████████████████████████████████████████| 83/83 [00:16<00:00, 4.94it/s]

train Loss: 0.1605 Acc: 0.9398

Processing batch 21: 100%|██████████████████████████████████████████████████████████████████████| 21/21 [00:01<00:00, 13.99it/s]

val Loss: 0.6605 Acc: 0.7543

Epoch complete in 18.3s

EarlyStopping counter: 5 out of 5

Early stopping

Training complete in 1m 50s

Best val Acc: 0.769800

\*\*TESTING\*\*

Loss used: CrossEntropyLoss()

Processing batch 860: 100%|██████████████████████████████████████████████████████████████████| 860/860 [00:05<00:00, 171.95it/s]

Test stats - Loss: 0.3896 Acc: 79.23%

Inference on Testset complete in 5.0s

Model saved at saved-models/DistillBertEmotion\_2025-05-07\_16-44-50\_trained\_testAcc=0.7923.pth

Training graph saved at figures/DistillBertEmotion\_losses&acc\_2025-05-07\_16-44-50\_testAcc=0.7923.png

Confusion Matrix saved at figures/DistillBertEmotion\_CM\_2025-05-07\_16-44-50\_testAcc=0.7923.png

Accuracy: 84.07%

Macro F1-score: 79.234%

Binary F1-score: 69.213%

MSE: 0.159

precision recall f1-score support

class 0 0.87 0.92 0.89 620

class 1 0.75 0.64 0.69 240

accuracy 0.84 860

macro avg 0.81 0.78 0.79 860

weighted avg 0.84 0.84 0.84 860说明了什么

从你使用 `DistillBertEmotion` 模型的训练和测试结果来看，模型在仇恨言论检测任务上的表现与基础 `DistillBert` 模型有所差异，以下是具体分析：

### 一、核心指标对比（与 `DistillBert` 对比）

| 指标 | DistillBert | DistillBertEmotion |

|---------------------|-------------------|-------------------|

| \*\*验证集最佳准确率\*\* | 77.40% | 76.98% |

| \*\*测试集准确率\*\* | 85.0% | 84.07% |

| \*\*类别1召回率\*\* | 65% | 64% |

| \*\*类别1 F1-score\*\* | 71% | 69% |

### 二、模型特性与任务适配性

1. \*\*预训练任务的影响\*\*：

- `DistillBertEmotion` 是针对 \*\*情感分析\*\* 预训练的模型，而当前任务是 \*\*仇恨言论检测\*\*。情感分析与仇恨言论有一定关联（如负面情感可能包含仇恨），但目标不同（情感分析关注情绪极性，仇恨言论关注攻击性内容）。

- 结果显示，情感预训练并未显著提升仇恨检测性能，甚至在类别1（仇恨言论）的召回率和F1-score上略有下降，说明 \*\*情感特征可能不是仇恨言论的强相关特征\*\*，或模型需要进一步微调以适配当前任务。

2. \*\*训练过程稳定性\*\*：

- 验证集准确率在 `Epoch 0` 达到最高（76.98%），随后逐渐下降，早停机制在 `Epoch 5` 触发，表明模型对验证集的拟合能力稍弱于基础 `DistillBert`，可能受限于情感预训练的特征偏向。

### 三、测试阶段具体表现

1. \*\*整体性能\*\*：

- \*\*测试准确率 84.07%\*\*：略低于基础模型（85%），但仍处于合理区间，说明模型具备一定泛化能力。

- \*\*Macro F1-score 79.23%\*\*：低于基础模型（80.27%），主要受类别1性能下降影响。

2. \*\*类别不平衡影响\*\*：

- \*\*类别0（非仇恨）\*\*：

precision 0.87、recall 0.92，表现稳定，说明模型对多数类的识别能力未受预训练任务明显影响。

- \*\*类别1（仇恨）\*\*：

precision 0.75（↓3%）、recall 0.64（↓1%），显示情感预训练可能导致模型对仇恨言论的判别边界偏移，更倾向于将部分仇恨样本误判为非仇恨（漏判增加）。

### 四、潜在原因分析

1. \*\*预训练模型适配不足\*\*：

- 情感预训练模型的特征空间与仇恨言论的语义空间存在差异，直接微调时未能有效提取任务关键特征（如攻击性词汇、语境）。

- 模型结构相同（仅预训练数据不同），但 `DistillBertEmotion` 的分类头可能更适配情感标签（如多类别情感），而非二分类仇恨标签。

2. \*\*数据与模型的特征错位\*\*：

- 仇恨言论检测依赖语义攻击性、歧视性词汇，而情感分析依赖情感极性（积极/消极），两者特征重叠度有限。例如，仇恨言论可能包含中性词汇（如歧视性标签），但情感模型对此类特征不敏感。

### 五、改进方向

1. \*\*模型选择\*\*：

- 优先使用通用领域预训练模型（如基础 `DistillBert`）或任务相关预训练模型（如针对有害语言的模型），而非情感专用模型。

- 尝试更大的预训练模型（如BERT-base）或任务定制模型（如 `bert-base-uncased-finetuned-sst-2-english` 等，但需验证适配性）。

2. \*\*微调策略优化\*\*：

- \*\*解冻更多预训练层\*\*：当前模型可能冻结了BERT主干（`freeze\_bert=True`），尝试解冻所有层（`freeze\_bert=False`），让模型学习任务特定特征。

- \*\*增加训练轮数\*\*：早停触发较早（Epoch 5），可适当提高耐心值（如 `patience\_es=10`），给模型更多微调时间。

3. \*\*数据增强与处理\*\*：

- \*\*领域特定数据增强\*\*：对仇恨言论样本添加对抗扰动（如替换同义词、插入攻击性词汇），强化模型对关键特征的敏感度。

- \*\*类别权重调整\*\*：在 `CrossEntropyLoss` 中增加类别1的权重（如 `weight=torch.tensor([1, 2])`），强制模型关注少数类。

### 六、总结

`DistillBertEmotion` 在仇恨言论检测任务上的表现略逊于基础 `DistillBert`，说明 \*\*预训练任务与目标任务的匹配度至关重要\*\*。当前结果验证了模型微调对预训练模型适配的重要性，后续可聚焦于：

1. 使用更贴合任务的预训练模型（如通用NLI或文本分类模型）；

2. 优化微调策略（如解冻更多层、调整学习率）；

3. 针对性处理类别不平衡，提升少数类检测能力。

通过以上调整，有望进一步提升模型对仇恨言论的检测效果，尤其是对少数类的召回率和F1-score。

# Gridsearch