# BasicLSTM

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# ### Example on our best BasicLSTM trained model root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# python -m src.evaluation.test\_save\_stats --model BasicLSTM --saved\_model\_path saved-models/BasicLSTM\_2025-05-07\_21-39-27\_trained\_testAcc=0.6931.pth --loss\_criterion bcelosswithlogits --only\_test 0 --stats\_label 1

model\_id: 2025-05-07\_21-39-27

Starting the test pipeline...

[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

loss\_criterion: bcelosswithlogits

saved-models/BasicLSTM\_2025-05-07\_21-39-27\_trained\_testAcc=0.6931.pth loaded.

Model BasicLSTM loaded on cuda

Start test and save stats

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██| 860/860 [00:01<00:00, 677.84it/s]

Test stats - Loss: 1.1637 Acc: 69.31%

Inference on Testset complete in 1.3s

Stats saved at stats-results/stats\_BasicLSTM\_2025-05-07\_21-39-27\_test\_bcelosswithlogits.csv

lowest\_stats\_df

original\_index ... loss

59 196 ... 0.001102

9 38 ... 0.001245

214 764 ... 0.001292

231 827 ... 0.001438

223 791 ... 0.001696

[5 rows x 6 columns]

highest\_stats\_df

original\_index ... loss

213 762 ... 9.840498

73 243 ... 9.639938

39 138 ... 9.549327

74 247 ... 9.436342

222 786 ... 9.423047

[5 rows x 6 columns]

说明了什么

从你给出的命令和输出信息来看，这是在使用 `BasicLSTM` 模型对仇恨言论检测任务进行测试，并且保存测试统计信息的过程。下面详细解读各部分输出所代表的含义：

### 1. 命令及参数说明

```bash

python -m src.evaluation.test\_save\_stats --model BasicLSTM --saved\_model\_path saved-models/BasicLSTM\_2025-05-07\_21-39-27\_trained\_testAcc=0.6931.pth --loss\_criterion bcelosswithlogits --only\_test 0 --stats\_label 1

```

- `--model BasicLSTM`：指定使用的模型为 `BasicLSTM`。

- `--saved\_model\_path saved-models/BasicLSTM\_2025-05-07\_21-39-27\_trained\_testAcc=0.6931.pth`：指定已训练好的模型的保存路径。

- `--loss\_criterion bcelosswithlogits`：指定使用的损失函数为 `BCEWithLogitsLoss`。

- `--only\_test 0`：表示不仅要进行测试，还要保存测试的统计信息。

- `--stats\_label 1`：指定只筛选真实标签为 1 的样本进行后续统计分析。

### 2. 测试前的准备阶段

```

model\_id: 2025-05-07\_21-39-27

Starting the test pipeline...

[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..

```

- 提取了模型的 ID 为 `2025-05-07\_21-39-27`。

- 启动测试流程，下载并检查 `nltk` 的停用词包，确认其为最新版本。

- 选择使用 `cuda` 设备（即 GPU）进行计算。

- 完成数据加载、格式化、划分训练集/验证集/测试集、创建字段对象、构建数据集对象、构建词汇表以及创建数据加载器等操作。

### 3. 模型加载阶段

```

model\_type: BasicLSTM

loss\_criterion: bcelosswithlogits

saved-models/BasicLSTM\_2025-05-07\_21-39-27\_trained\_testAcc=0.6931.pth loaded.

Model BasicLSTM loaded on cuda

```

- 确认使用的模型类型为 `BasicLSTM`，损失函数为 `BCEWithLogitsLoss`。

- 成功加载指定路径下的 `BasicLSTM` 模型，并将其部署到 `cuda` 设备上。

### 4. 测试阶段

```

Start test and save stats

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██| 860/860 [00:01<00:00, 677.84it/s]

Test stats - Loss: 1.1637 Acc: 69.31%

Inference on Testset complete in 1.3s

```

- 开始测试并保存统计信息。

- 明确使用的损失函数为 `BCEWithLogitsLoss`。

- 处理了 860 个批次的数据，处理速度约为 677.84 个批次每秒。

- 测试集上的损失值为 1.1637，准确率为 69.31%，整个测试过程耗时 1.3 秒。

### 5. 统计信息保存阶段

```

Stats saved at stats-results/stats\_BasicLSTM\_2025-05-07\_21-39-27\_test\_bcelosswithlogits.csv

```

将测试的统计信息保存到 `stats-results/stats\_BasicLSTM\_2025-05-07\_21-39-27\_test\_bcelosswithlogits.csv` 文件中。

### 6. 统计信息分析阶段

```

lowest\_stats\_df

original\_index ... loss

59 196 ... 0.001102

9 38 ... 0.001245

214 764 ... 0.001292

231 827 ... 0.001438

223 791 ... 0.001696

[5 rows x 6 columns]

highest\_stats\_df

original\_index ... loss

213 762 ... 9.840498

73 243 ... 9.639938

39 138 ... 9.549327

74 247 ... 9.436342

222 786 ... 9.423047

[5 rows x 6 columns]

```

- `lowest\_stats\_df`：展示了真实标签为 1 的样本中，损失值最低的前 5 个样本的原始索引和损失值。这些样本可能是模型预测最准确的样本。

- `highest\_stats\_df`：展示了真实标签为 1 的样本中，损失值最高的前 5 个样本的原始索引和损失值。这些样本可能是模型预测最不准确的样本。

总体而言，这个输出展示了 `BasicLSTM` 模型在测试集上的性能表现，包括测试过程、损失值、准确率，以及保存的统计信息和对统计信息的初步分析。这些信息可以帮助你评估模型的性能，找出模型预测不准确的样本，进而对模型进行优化。

# BiLSTM

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# python -m src.evaluation.test\_save\_stats --model BiLSTM --saved\_model\_path saved-models/BiLSTM\_2025-05-07\_21-40-13\_trained\_testAcc=0.6905.pth --loss\_criterion bcelosswithlogits --only\_test 0 --stats\_label 1

model\_id: 2025-05-07\_21-40-13

Starting the test pipeline...

[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

loss\_criterion: bcelosswithlogits

saved-models/BiLSTM\_2025-05-07\_21-40-13\_trained\_testAcc=0.6905.pth loaded.

Model BiLSTM loaded on cuda

Start test and save stats

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████████████████████████████████████████████| 860/860 [00:01<00:00, 476.32it/s]

Test stats - Loss: 0.5242 Acc: 69.05%

Inference on Testset complete in 1.8s

Stats saved at stats-results/stats\_BiLSTM\_2025-05-07\_21-40-13\_test\_bcelosswithlogits.csv

lowest\_stats\_df

original\_index text ... prob loss

198 701 @user nigga stupid trash nt play play bitch : ... ... 0.980724 0.019464

214 764 disarmthem antifa please one stupid enough . u... ... 0.977497 0.022760

60 201 @user <unk> attacks idiot <unk> ! get worse mu... ... 0.975646 0.024655

181 632 anytime <unk> <unk> laughing ’s remembered dum... ... 0.975357 0.024952

133 443 $ b album definitely worth fucking wait ... 0.975330 0.024979

[5 rows x 6 columns]

highest\_stats\_df

original\_index text ... prob loss

141 475 @user diversity works joining new group want p... ... 0.042594 3.156034

158 531 @user 'cause everyone knows real leader falls ... ... 0.050881 2.978264

85 279 <unk> funds <unk> antifa url ... 0.062995 2.764708

37 135 boycottnfl .@user dead . @user @user url ... 0.067231 2.699626

94 313 @user @user @user disingenuous person senate .... ... 0.069689 2.663709

[5 rows x 6 columns]

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# 说明了什么

从你给出的命令执行输出信息来看，代码成功地对 `BiLSTM` 模型进行了测试，并且保存了相关统计信息。下面为你详细解读这些输出信息：

### 1. 命令执行流程概述

- \*\*参数设置\*\*：你运行的命令指定了模型类型为 `BiLSTM`，加载的预训练模型路径为 `saved-models/BiLSTM\_2025-05-07\_21-40-13\_trained\_testAcc=0.6905.pth`，损失函数为 `bcelosswithlogits`，`--only\_test` 设置为 `0` 表示要进行测试并保存统计信息，`--stats\_label` 设置为 `1` 表示只考虑真实标签为 `1` 的样本。

- \*\*数据处理与模型加载\*\*：代码完成了数据的加载、格式化、划分（训练集、验证集、测试集），创建了字段对象、数据集对象和词汇表，还创建了数据加载器。接着，成功加载了指定的 `BiLSTM` 模型到 `cuda` 设备上。

- \*\*测试与统计信息保存\*\*：代码对测试集进行了推理，计算了损失和准确率，并将每个样本的详细统计信息保存到了 CSV 文件中。最后，输出了损失最低和最高的前 5 个样本的信息。

### 2. 详细输出解读

#### 模型加载与数据处理阶段

```plaintext

model\_id: 2025-05-07\_21-40-13

Starting the test pipeline...

[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\_id`：标识了所使用的预训练模型的版本。

- `Starting the test pipeline...`：表明测试流程开始。

- `nltk\_data` 相关信息：显示了 `nltk` 停用词包的下载情况，这里显示该包已是最新版本。

- `Device: cuda`：说明模型将在 GPU 上运行。

- 后续几行信息表明数据处理的各个步骤已成功完成。

#### 模型与损失函数信息

```plaintext

model\_type: BiLSTM

loss\_criterion: bcelosswithlogits

saved-models/BiLSTM\_2025-05-07\_21-40-13\_trained\_testAcc=0.6905.pth loaded.

Model BiLSTM loaded on cuda

```

- 明确了使用的模型类型和损失函数。

- 确认指定的预训练模型已成功加载到 GPU 上。

#### 测试阶段

```plaintext

Start test and save stats

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████████████████████████████████████████████| 860/860 [00:01<00:00, 476.32it/s]

Test stats - Loss: 0.5242 Acc: 69.05%

Inference on Testset complete in 1.8s

```

- `Start test and save stats`：表明开始进行测试并保存统计信息。

- `Loss used: BCEWithLogitsLoss()`：再次确认使用的损失函数。

- 进度条显示了测试过程中批次的处理进度。

- `Test stats - Loss: 0.5242 Acc: 69.05%`：给出了测试集上的损失和准确率，说明该 `BiLSTM` 模型在测试集上的表现，准确率为 69.05%，损失为 0.5242。

- `Inference on Testset complete in 1.8s`：显示测试集推理所用的时间。

#### 统计信息保存

```plaintext

Stats saved at stats-results/stats\_BiLSTM\_2025-05-07\_21-40-13\_test\_bcelosswithlogits.csv

```

- 表明统计信息已成功保存到指定的 CSV 文件中，该文件包含了每个样本的原始索引、文本、真实标签、预测标签、概率和损失等信息。

#### 损失最低和最高的样本信息

```plaintext

lowest\_stats\_df

original\_index text ... prob loss

198 701 @user nigga stupid trash nt play play bitch : ... ... 0.980724 0.019464

214 764 disarmthem antifa please one stupid enough . u... ... 0.977497 0.022760

60 201 @user <unk> attacks idiot <unk> ! get worse mu... ... 0.975646 0.024655

181 632 anytime <unk> <unk> laughing ’s remembered dum... ... 0.975357 0.024952

133 443 $ b album definitely worth fucking wait ... 0.975330 0.024979

[5 rows x 6 columns]

highest\_stats\_df

original\_index text ... prob loss

141 475 @user diversity works joining new group want p... ... 0.042594 3.156034

158 531 @user 'cause everyone knows real leader falls ... ... 0.050881 2.978264

85 279 <unk> funds <unk> antifa url ... 0.062995 2.764708

37 135 boycottnfl .@user dead . @user @user url ... 0.067231 2.699626

94 313 @user @user @user disingenuous person senate .... ... 0.069689 2.663709

[5 rows x 6 columns]

```

- `lowest\_stats\_df`：展示了损失最低的前 5 个样本，这些样本的预测概率较高（接近 1），损失较低，说明模型对这些样本的预测较为准确。

- `highest\_stats\_df`：展示了损失最高的前 5 个样本，这些样本的预测概率较低（接近 0），损失较高，说明模型对这些样本的预测存在较大误差。

### 3. 总结

- 该 `BiLSTM` 模型在测试集上的准确率为 69.05%，可以作为该模型在仇恨言论检测任务上的性能参考。

- 通过查看损失最低和最高的样本，可以进一步分析模型的优缺点，例如模型在哪些类型的文本上表现较好，在哪些类型的文本上容易出错，从而为模型的改进提供方向。

- 保存的统计信息 CSV 文件可以用于后续的详细分析和可视化。

# HybridCNNLSTM

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# python -m src.evaluation.test\_save\_stats --model HybridCNNLSTM --saved\_model\_path saved-models/HybridCNNLSTM\_2025-05-07\_21-40-59\_trained\_testAcc=0.6645.pth --loss\_criterion bcelosswithlogits --only\_test 0 --stats\_label 1model\_id: 2025-05-07\_21-40-59

Starting the test pipeline...

[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

loss\_criterion: bcelosswithlogits

saved-models/HybridCNNLSTM\_2025-05-07\_21-40-59\_trained\_testAcc=0.6645.pth loaded.

Model HybridCNNLSTM loaded on cuda

Start test and save stats

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████████████████████████████████████████████| 860/860 [00:02<00:00, 406.30it/s]

Test stats - Loss: 0.6425 Acc: 66.45%

Inference on Testset complete in 2.1s

Stats saved at stats-results/stats\_HybridCNNLSTM\_2025-05-07\_21-40-59\_test\_bcelosswithlogits.csv

lowest\_stats\_df

original\_index text ... prob loss

60 201 @user <unk> attacks idiot <unk> ! get worse mu... ... 0.999983 0.000017

198 701 @user nigga stupid trash nt play play bitch : ... ... 0.999749 0.000251

127 428 anyone finds <unk> 's behavior funny fuck die ... 0.999256 0.000744

163 559 <unk> liar fat ugly <unk> snowflake sold get c... ... 0.999189 0.000811

9 38 <unk> ? serious ? ! holy shit , please fucking... ... 0.998439 0.001563

[5 rows x 6 columns]

highest\_stats\_df

original\_index text ... prob loss

141 475 @user diversity works joining new group want p... ... 0.000847 7.074364

118 400 <unk> actually question people want <unk> win ... ... 0.002392 6.035609

155 512 america ... tear wall ! url tcot <unk> trump <... ... 0.002842 5.863393

108 357 4 . ( part 2 ) n't think hard come <unk> . 're... ... 0.003088 5.780097

58 189 <unk> spain <unk> . spain fakedemocracy @user ... ... 0.003250 5.728983

[5 rows x 6 columns]

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey#

从你提供的命令执行输出信息来看，代码完成了对 `HybridCNNLSTM` 模型在测试集上的评估，并保存了相关统计信息。下面详细分析输出内容所说明的情况：

### 1. 环境和数据准备

```

Starting the test pipeline...

[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..

```

- \*\*环境准备\*\*：代码启动了测试流程，尝试下载 `nltk` 的 `stopwords` 包，但该包已为最新版本。同时，代码检测到可用的 `cuda` 设备，因此将使用 GPU 进行计算。

- \*\*数据处理\*\*：代码完成了数据的加载、格式化、划分（训练集、验证集和测试集），创建了 `field` 对象和数据集对象，构建了词汇表，并创建了数据加载器。

### 2. 模型和损失函数信息

```

model\_type: HybridCNNLSTM

loss\_criterion: bcelosswithlogits

```

这表明使用的模型是 `HybridCNNLSTM`，损失函数是 `BCEWithLogitsLoss`，即二元交叉熵损失函数结合了 sigmoid 激活函数。

### 3. 模型加载

```

saved-models/HybridCNNLSTM\_2025-05-07\_21-40-59\_trained\_testAcc=0.6645.pth loaded.

Model HybridCNNLSTM loaded on cuda

```

代码成功加载了预训练的 `HybridCNNLSTM` 模型，并将其移动到 `cuda` 设备上。

### 4. 测试过程和结果

```

Start test and save stats

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████████████████████████████████████████████| 860/860 [00:02<00:00, 406.30it/s]

Test stats - Loss: 0.6425 Acc: 66.45%

Inference on Testset complete in 2.1s

```

- \*\*测试启动\*\*：代码开始对测试集进行评估，并保存相关统计信息。

- \*\*损失函数\*\*：明确使用 `BCEWithLogitsLoss` 作为损失函数。

- \*\*测试进度\*\*：代码处理了 860 个批次的数据，处理速度约为 406.30 个批次/秒，总耗时 2.1 秒。

- \*\*测试结果\*\*：模型在测试集上的损失为 0.6425，准确率为 66.45%。

### 5. 统计信息保存

```

Stats saved at stats-results/stats\_HybridCNNLSTM\_2025-05-07\_21-40-59\_test\_bcelosswithlogits.csv

```

代码将测试结果的统计信息保存到了指定的 CSV 文件中。

### 6. 最低和最高指标样本

```

lowest\_stats\_df

original\_index text ... prob loss

60 201 @user <unk> attacks idiot <unk> ! get worse mu... ... 0.999983 0.000017

...

highest\_stats\_df

original\_index text ... prob loss

141 475 @user diversity works joining new group want p... ... 0.000847 7.074364

...

```

- \*\*最低损失样本\*\*：`lowest\_stats\_df` 展示了损失最低的 5 个样本，这些样本的预测概率接近 1，损失接近 0，说明模型对这些样本的预测非常准确。

- \*\*最高损失样本\*\*：`highest\_stats\_df` 展示了损失最高的 5 个样本，这些样本的预测概率接近 0，损失较大，说明模型对这些样本的预测效果较差。

### 总结

输出结果表明，`HybridCNNLSTM` 模型在测试集上的准确率为 66.45%，并且代码成功保存了测试结果的统计信息。通过分析最低和最高损失的样本，可以进一步了解模型在哪些样本上表现较好或较差，有助于后续的模型优化和调试。

# PyramidCNN

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# rm stats-results/stats\_PyramidCNN\_2025-05-08\_01-41-21\_test\_bcelosswithlogits.csv

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# python -m src.evaluation.test\_save\_stats \

> --model PyramidCNN \

> --saved\_model\_path saved-models/PyramidCNN\_2025-05-08\_01-41-21\_trained\_testAcc=0.6916.pth \

> --loss\_criterion bcelosswithlogits \

> --pyramid "256,256" \

> --fcs "128,256,1" \

> --context\_size 2 \

> --only\_test 0 \

> --stats\_label 3

model\_id: 2025-05-08\_01-41-21

Starting the test pipeline...

[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

loss\_criterion: bcelosswithlogits

Context size is 2

saved-models/PyramidCNN\_2025-05-08\_01-41-21\_trained\_testAcc=0.6916.pth 已加载，兼容模式。

Model PyramidCNN loaded on cuda

Start test and save stats

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████| 860/860 [00:01<00:00, 439.57it/s]

Test stats - Loss: 0.6744 Acc: 58.02%

Inference on Testset complete in 2.0s

Stats saved at stats-results/stats\_PyramidCNN\_2025-05-08\_01-41-21\_test\_bcelosswithlogits.csv

lowest\_stats\_df

Empty DataFrame

Columns: [original\_index, text, true\_label, pred\_label, prob, loss]

Index: []

highest\_stats\_df

Empty DataFrame

Columns: [original\_index, text, true\_label, pred\_label, prob, loss]

Index: []

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey#

这段输出信息展示了对 `PyramidCNN` 模型进行测试并保存统计信息的完整流程，下面详细分析每个部分所说明的情况：

### 1. 命令执行与环境准备

```

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# rm stats-results/stats\_PyramidCNN\_2025-05-08\_01-41-21\_test\_bcelosswithlogits.csv

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# python -m src.evaluation.test\_save\_stats \

> --model PyramidCNN \

> --saved\_model\_path saved-models/PyramidCNN\_2025-05-08\_01-41-21\_trained\_testAcc=0.6916.pth \

> --loss\_criterion bcelosswithlogits \

> --pyramid "256,256" \

> --fcs "128,256,1" \

> --context\_size 2 \

> --only\_test 0 \

> --stats\_label 3

```

- 首先删除了之前保存的统计信息 CSV 文件。

- 接着执行 `src.evaluation.test\_save\_stats` 模块，指定使用 `PyramidCNN` 模型，并加载预训练的模型权重文件。同时设置了损失函数、金字塔结构参数、全连接层参数、上下文大小等超参数。`--only\_test 0` 表示不仅要进行测试，还要保存统计信息；`--stats\_label 3` 表示只关注真实标签为 3 的样本。

### 2. 数据处理与模型加载

```

model\_id: 2025-05-08\_01-41-21

Starting the test pipeline...

[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

loss\_criterion: bcelosswithlogits

Context size is 2

saved-models/PyramidCNN\_2025-05-08\_01-41-21\_trained\_testAcc=0.6916.pth 已加载，兼容模式。

Model PyramidCNN loaded on cuda

```

- 确定了模型的 ID。

- 启动测试流程，下载 `nltk` 的 `stopwords` 包（如果需要），并检测到可用的 `cuda` 设备，将使用 GPU 进行计算。

- 完成数据的加载、格式化、划分（训练集、验证集和测试集），创建了 `field` 对象和数据集对象，构建了词汇表，并创建了数据加载器。

- 确认了模型类型和损失函数，设置了上下文大小为 2。

- 成功加载预训练的 `PyramidCNN` 模型，并将其移动到 `cuda` 设备上。

### 3. 测试过程与结果

```

Start test and save stats

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████| 860/860 [00:01<00:00, 439.57it/s]

Test stats - Loss: 0.6744 Acc: 58.02%

Inference on Testset complete in 2.0s

```

- 开始测试并保存统计信息。

- 明确使用 `BCEWithLogitsLoss` 作为损失函数。

- 处理了 860 个批次的数据，处理速度约为 439.57 个批次/秒，总耗时 2.0 秒。

- 模型在测试集上的损失为 0.6744，准确率为 58.02%。

### 4. 统计信息保存

```

Stats saved at stats-results/stats\_PyramidCNN\_2025-05-08\_01-41-21\_test\_bcelosswithlogits.csv

```

代码将测试结果的统计信息保存到了指定的 CSV 文件中。

### 5. 最低和最高指标样本

```

lowest\_stats\_df

Empty DataFrame

Columns: [original\_index, text, true\_label, pred\_label, prob, loss]

Index: []

highest\_stats\_df

Empty DataFrame

Columns: [original\_index, text, true\_label, pred\_label, prob, loss]

Index: []

```

- `lowest\_stats\_df` 和 `highest\_stats\_df` 均为空，这表明在测试集中没有真实标签为 3 的样本。可能是测试集中根本不存在真实标签为 3 的数据，或者数据预处理过程中出现了问题，导致这类样本被过滤掉了。

### 总结

输出结果表明，`PyramidCNN` 模型在测试集上的准确率为 58.02%，并且代码成功保存了测试结果的统计信息。但由于指定的 `stats\_label` 为 3，而测试集中可能不存在真实标签为 3 的样本，因此无法找到损失最低和最高的样本。你可以检查数据集，确认是否存在真实标签为 3 的样本，或者调整 `stats\_label` 的值以获取有效的统计信息。

# HybridLSTMCNN

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# python -m src.evaluation.test\_save\_stats --model HybridLSTMCNN --saved\_model\_path saved-models/HybridLSTMCNN\_2025-05-07\_21-46-36\_trained\_testAcc=0.6726.pth --loss\_criterion bcelosswithlogits --only\_test 0 --stats\_label 1 --fix\_length 20

model\_id: 2025-05-07\_21-46-36

Starting the test pipeline...

[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

loss\_criterion: bcelosswithlogits

saved-models/HybridLSTMCNN\_2025-05-07\_21-46-36\_trained\_testAcc=0.6726.pth 已加载，兼容模式。

Model HybridLSTMCNN loaded on cuda

Start test and save stats

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████| 860/860 [00:01<00:00, 443.76it/s]

Test stats - Loss: 0.8286 Acc: 67.26%

Inference on Testset complete in 1.9s

Stats saved at stats-results/stats\_HybridLSTMCNN\_2025-05-07\_21-46-36\_test\_bcelosswithlogits.csv

lowest\_stats\_df

original\_index ... loss

77 259 ... 1.192093e-07

151 505 ... 3.576278e-07

198 701 ... 2.145765e-06

223 791 ... 2.741810e-06

115 377 ... 4.410734e-06

[5 rows x 6 columns]

highest\_stats\_df

original\_index ... loss

86 281 ... 9.200315

220 778 ... 7.256581

219 775 ... 6.916979

36 130 ... 6.741055

184 638 ... 6.130057

[5 rows x 6 columns]

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey#

这段输出展示了使用 `HybridLSTMCNN` 模型对测试集进行评估并保存统计信息的过程和结果，下面详细解读每个部分：

### 1. 命令执行与环境准备

```

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# python -m src.evaluation.test\_save\_stats --model HybridLSTMCNN --saved\_model\_path saved-models/HybridLSTMCNN\_2025-05-07\_21-46-36\_trained\_testAcc=0.6726.pth --loss\_criterion bcelosswithlogits --only\_test 0 --stats\_label 1 --fix\_length 20

```

- 执行 `src.evaluation.test\_save\_stats` 模块，指定使用 `HybridLSTMCNN` 模型，并加载预训练的模型权重文件 `saved-models/HybridLSTMCNN\_2025-05-07\_21-46-36\_trained\_testAcc=0.6726.pth`。

- 使用 `BCEWithLogitsLoss` 作为损失函数。

- `--only\_test 0` 表示不仅要进行测试，还要保存统计信息。

- `--stats\_label 1` 表示只关注真实标签为 1 的样本。

- `--fix\_length 20` 解决了之前 `fix\_length` 为 `None` 导致的类型错误问题。

### 2. 数据处理与模型加载

```

model\_id: 2025-05-07\_21-46-36

Starting the test pipeline...

[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

loss\_criterion: bcelosswithlogits

saved-models/HybridLSTMCNN\_2025-05-07\_21-46-36\_trained\_testAcc=0.6726.pth 已加载，兼容模式。

Model HybridLSTMCNN loaded on cuda

```

- 确定了模型的 ID 为 `2025-05-07\_21-46-36`。

- 启动测试流程，下载 `nltk` 的 `stopwords` 包（如果需要），并检测到可用的 `cuda` 设备，将使用 GPU 进行计算。

- 完成数据的加载、格式化、划分（训练集、验证集和测试集），创建了 `field` 对象和数据集对象，构建了词汇表，并创建了数据加载器。

- 成功加载预训练的 `HybridLSTMCNN` 模型，并将其移动到 `cuda` 设备上。

### 3. 测试过程与结果

```

Start test and save stats

\*\*TESTING\*\*

Loss used: BCEWithLogitsLoss()

Processing batch 860: 100%|██████| 860/860 [00:01<00:00, 443.76it/s]

Test stats - Loss: 0.8286 Acc: 67.26%

Inference on Testset complete in 1.9s

```

- 开始测试并保存统计信息。

- 明确使用 `BCEWithLogitsLoss` 作为损失函数。

- 处理了 860 个批次的数据，处理速度约为 443.76 个批次/秒，总耗时 1.9 秒。

- 模型在测试集上的损失为 0.8286，准确率为 67.26%。

### 4. 统计信息保存

```

Stats saved at stats-results/stats\_HybridLSTMCNN\_2025-05-07\_21-46-36\_test\_bcelosswithlogits.csv

```

代码将测试结果的统计信息保存到了指定的 CSV 文件中。

### 5. 最低和最高指标样本

```

lowest\_stats\_df

original\_index ... loss

77 259 ... 1.192093e-07

151 505 ... 3.576278e-07

198 701 ... 2.145765e-06

223 791 ... 2.741810e-06

115 377 ... 4.410734e-06

[5 rows x 6 columns]

highest\_stats\_df

original\_index ... loss

86 281 ... 9.200315

220 778 ... 7.256581

219 775 ... 6.916979

36 130 ... 6.741055

184 638 ... 6.130057

[5 rows x 6 columns]

```

- `lowest\_stats\_df` 展示了损失最低的 5 个样本的信息，包括原始索引和损失值。这些样本可能是模型预测最准确的样本，模型对这些样本的输出与真实标签非常接近，因此损失值很小。

- `highest\_stats\_df` 展示了损失最高的 5 个样本的信息，包括原始索引和损失值。这些样本可能是模型预测最不准确的样本，模型对这些样本的输出与真实标签差异较大，因此损失值很大。

### 总结

这段输出表明，通过添加 `--fix\_length 20` 参数，成功解决了之前的类型错误问题，`HybridLSTMCNN` 模型在测试集上的准确率为 67.26%。同时，代码保存了测试结果的统计信息，并展示了损失最低和最高的 5 个样本的信息，这些信息可以帮助你进一步分析模型的性能，找出模型容易出错的样本，从而进行针对性的改进。

# DistillBert

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# python -m src.evaluation.test\_save\_stats --model DistillBert --saved\_model\_path saved-models/DistillBert\_2025-05-07\_22-05-24\_trained\_testAcc=0.7993.pth --loss\_criterion crossentropy

model\_id: 2025-05-07\_22-05-24

Starting the test pipeline...

[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

loss\_criterion: crossentropy

saved-models/DistillBert\_2025-05-07\_22-05-24\_trained\_testAcc=0.7993.pth 已加载，兼容模式。

Model DistillBert loaded on cuda

Start test and save stats

\*\*TESTING\*\*

Loss used: CrossEntropyLoss()

Processing batch 860: 100%|███████████████████████████████████████████████████████████████████████| 860/860 [00:05<00:00, 168.55it/s]

Test stats - Loss: 0.3691 Acc: 79.93%

Inference on Testset complete in 5.1s

Stats saved at stats-results/stats\_DistillBert\_2025-05-07\_22-05-24\_test\_crossentropy.csv

lowest\_stats\_df

original\_index text true\_label pred\_label prob loss

156 225 [CLS] wcw to @ user i mean hey, she is a great... 0 0 0.027348 0.027729

557 777 [CLS] @ user watch ielts liz.. she is amazing ... 0 0 0.028350 0.028760

31 44 [CLS] weloveseungcheol @ user i am happy and p... 0 0 0.028887 0.029312

513 715 [CLS] feelsgoodgirl @ user @ user you should b... 0 0 0.029447 0.029889

79 113 [CLS] supernaturaltnt themanwhowouldbeking omg... 0 0 0.031511 0.032019

highest\_stats\_df

original\_index text true\_label pred\_label prob loss

42 58 [CLS] are you fucking serious? url [SEP] 0 1 0.903914 2.342513

514 716 [CLS] @ user i guess that ’ s where swamp ass ... 0 1 0.876302 2.089910

482 674 [CLS] alex jones be smokin be dicks out here b... 0 1 0.860111 1.966909

437 616 [CLS] @ user @ user @ user she is just nasty [... 0 1 0.850631 1.901334

310 444 [CLS] 3 days before bbc radio 2 ’ s festival i... 0 1 0.765153 1.448822

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey#

这段输出展示了使用 `DistillBert` 模型在测试集上进行评估并保存统计信息的详细过程和结果。下面是对各个部分的详细解读：

### 1. 命令执行与环境准备

```bash

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# python -m src.evaluation.test\_save\_stats --model DistillBert --saved\_model\_path saved-models/DistillBert\_2025-05-07\_22-05-24\_trained\_testAcc=0.7993.pth --loss\_criterion crossentropy

```

- 执行 `src.evaluation.test\_save\_stats` 模块，指定使用 `DistillBert` 模型，并加载预训练的模型权重文件 `saved-models/DistillBert\_2025-05-07\_22-05-24\_trained\_testAcc=0.7993.pth`。

- 使用 `CrossEntropyLoss` 作为损失函数。

### 2. 数据处理与模型加载

```plaintext

model\_id: 2025-05-07\_22-05-24

Starting the test pipeline...

[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

loss\_criterion: crossentropy

saved-models/DistillBert\_2025-05-07\_22-05-24\_trained\_testAcc=0.7993.pth 已加载，兼容模式。

Model DistillBert loaded on cuda

```

- 确定了模型的 ID 为 `2025-05-07\_22-05-24`。

- 启动测试流程，下载 `nltk` 的 `stopwords` 包（如果需要），并检测到可用的 `cuda` 设备，将使用 GPU 进行计算。

- 完成数据的加载、格式化、划分（训练集、验证集和测试集），创建了 `field` 对象和数据集对象，构建了词汇表，并创建了数据加载器。

- 成功加载预训练的 `DistillBert` 模型，并将其移动到 `cuda` 设备上。

### 3. 测试过程与结果

```plaintext

Start test and save stats

\*\*TESTING\*\*

Loss used: CrossEntropyLoss()

Processing batch 860: 100%|███████████████████████████████████████████████████████████████████████| 860/860 [00:05<00:00, 168.55it/s]

Test stats - Loss: 0.3691 Acc: 79.93%

Inference on Testset complete in 5.1s

```

- 开始测试并保存统计信息。

- 明确使用 `CrossEntropyLoss` 作为损失函数。

- 处理了 860 个批次的数据，处理速度约为 168.55 个批次/秒，总耗时 5.1 秒。

- 模型在测试集上的损失为 0.3691，准确率为 79.93%，这表明模型在测试集上表现良好。

### 4. 统计信息保存

```plaintext

Stats saved at stats-results/stats\_DistillBert\_2025-05-07\_22-05-24\_test\_crossentropy.csv

```

将测试结果的统计信息保存到了指定的 CSV 文件中，方便后续分析。

### 5. 最低和最高指标样本

```plaintext

lowest\_stats\_df

original\_index text true\_label pred\_label prob loss

156 225 [CLS] wcw to @ user i mean hey, she is a great... 0 0 0.027348 0.027729

557 777 [CLS] @ user watch ielts liz.. she is amazing ... 0 0 0.028350 0.028760

31 44 [CLS] weloveseungcheol @ user i am happy and p... 0 0 0.028887 0.029312

513 715 [CLS] feelsgoodgirl @ user @ user you should b... 0 0 0.029447 0.029889

79 113 [CLS] supernaturaltnt themanwhowouldbeking omg... 0 0 0.031511 0.032019

highest\_stats\_df

original\_index text true\_label pred\_label prob loss

42 58 [CLS] are you fucking serious? url [SEP] 0 1 0.903914 2.342513

514 716 [CLS] @ user i guess that ’ s where swamp ass ... 0 1 0.876302 2.089910

482 674 [CLS] alex jones be smokin be dicks out here b... 0 1 0.860111 1.966909

437 616 [CLS] @ user @ user @ user she is just nasty [... 0 1 0.850631 1.901334

310 444 [CLS] 3 days before bbc radio 2 ’ s festival i... 0 1 0.765153 1.448822

```

- `lowest\_stats\_df` 展示了损失最低的 5 个样本的信息，包括原始索引、文本内容、真实标签、预测标签、预测概率和损失值。这些样本的真实标签和预测标签一致，且损失值很小，说明模型对这些样本的预测非常准确。

- `highest\_stats\_df` 展示了损失最高的 5 个样本的信息。这些样本的真实标签为 0（可能表示非仇恨言论），但模型预测为 1（可能表示仇恨言论），预测概率较高，损失值也较大，说明模型在这些样本上出现了错误预测。这些错误可能是由于文本中的一些词汇或表述容易让模型产生误判，例如包含一些不文明用语但实际上并非仇恨言论的文本。

### 总结

这段输出表明 `DistillBert` 模型在测试集上取得了 79.93% 的准确率，表现较好。通过分析损失最低和最高的样本，可以进一步了解模型的优势和不足，有助于后续对模型进行优化和改进，例如对容易误判的样本进行更深入的分析，调整模型的参数或进行更多的数据增强等操作。

# DistillBertEmotion

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# python -m src.evaluation.test\_save\_stats --model DistillBertEmotion --saved\_model\_path saved-models/DistillBertEmotion\_2025-05-07\_22-03-00\_trained\_testAcc=0.7935.pth --loss\_criterion crossentropy

model\_id: 2025-05-07\_22-03-00

Starting the test pipeline...

[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

loss\_criterion: crossentropy

saved-models/DistillBertEmotion\_2025-05-07\_22-03-00\_trained\_testAcc=0.7935.pth 已加载，兼容模式。

Model DistillBertEmotion loaded on cuda

Start test and save stats

\*\*TESTING\*\*

Loss used: CrossEntropyLoss()

Processing batch 860: 100%|███████████████████████████████████████████████████████████████████████| 860/860 [00:04<00:00, 197.66it/s]

Test stats - Loss: 0.3881 Acc: 79.35%

Inference on Testset complete in 4.4s

Stats saved at stats-results/stats\_DistillBertEmotion\_2025-05-07\_22-03-00\_test\_crossentropy.csv

lowest\_stats\_df

original\_index text true\_label pred\_label prob loss

513 715 2228 5386 <UNK> <UNK> 3693 18132 3693 18132 49... 0 0 0.053552 0.055039

453 636 2228 3693 18132 3693 18132 7486 4887 4281 4000 0 0 0.054818 0.056377

233 337 2228 2906 <UNK> 6714 <UNK> 25811 5312 4953 534... 0 0 0.057254 0.058958

37 53 2228 24716 24955 5843 12581 2394 4512 5843 470... 0 0 0.059632 0.061484

354 506 2228 3693 18132 22852 4419 18155 3569 22852 64... 0 0 0.060784 0.062710

highest\_stats\_df

original\_index text true\_label pred\_label prob loss

425 599 2228 6181 <UNK> 4281 25022 <UNK> 5358 16987 <U... 0 1 0.902435 2.327241

502 700 2228 5091 5491 5858 5053 4281 5627 3810 3782 2... 0 1 0.886540 2.176303

514 716 2228 3693 18132 5843 26831 4512 3560 18155 151... 0 1 0.880616 2.125407

42 58 2228 5341 4953 19527 9694 3533 <UNK> 5671 4000 0 1 0.874697 2.077023

482 674 2228 3603 2827 5247 <UNK> <UNK> 19006 5247 946... 0 1 0.869784 2.038563

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey#

从你提供的命令执行输出信息来看，程序成功完成了对 `DistillBertEmotion` 模型的测试，并保存了相关统计信息。下面详细解读这些输出说明了什么：

### 1. 命令执行与初始化信息

```

root@autodl-container-11d445a847-184a60d3:~/hate/hate-speech-detection-survey# python -m src.evaluation.test\_save\_stats --model DistillBertEmotion --saved\_model\_path saved-models/DistillBertEmotion\_2025-05-07\_22-03-00\_trained\_testAcc=0.7935.pth --loss\_criterion crossentropy

model\_id: 2025-05-07\_22-03-00

Starting the test pipeline...

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

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

Device: cuda

```

- \*\*命令执行\*\*：使用 `python -m` 运行 `src.evaluation.test\_save\_stats` 模块，指定模型为 `DistillBertEmotion`，加载预训练模型 `saved-models/DistillBertEmotion\_2025-05-07\_22-03-00\_trained\_testAcc=0.7935.pth`，并使用 `crossentropy` 作为损失函数。

- \*\*模型 ID\*\*：从保存的模型路径中提取出模型 ID 为 `2025-05-07\_22-03-00`。

- \*\*测试流程启动\*\*：程序开始执行测试流程。

- \*\*NLTK 数据\*\*：尝试下载 `stopwords` 数据集，但发现已经是最新版本。

- \*\*设备信息\*\*：程序检测到可用的 CUDA 设备，并将使用 GPU 进行计算。

### 2. 数据处理与模型加载信息

```

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

loss\_criterion: crossentropy

saved-models/DistillBertEmotion\_2025-05-07\_22-03-00\_trained\_testAcc=0.7935.pth 已加载，兼容模式。

Model DistillBertEmotion loaded on cuda

Start test and save stats

```

- \*\*数据处理\*\*：程序完成了数据的加载、格式化、划分训练集/验证集/测试集，创建了字段对象和数据集对象，并构建了词汇表，最后创建了数据加载器。

- \*\*模型信息\*\*：确认使用的模型类型为 `DistillBertEmotion`，损失函数为 `crossentropy`。

- \*\*模型加载\*\*：成功加载预训练模型，并将其移动到 CUDA 设备上。

- \*\*测试开始\*\*：程序开始执行测试并保存统计信息。

### 3. 测试过程与结果信息

```

\*\*TESTING\*\*

Loss used: CrossEntropyLoss()

Processing batch 860: 100%|███████████████████████████████████████████████████████████████████████| 860/860 [00:04<00:00, 197.66it/s]

Test stats - Loss: 0.3881 Acc: 79.35%

Inference on Testset complete in 4.4s

```

- \*\*测试阶段\*\*：程序进入测试阶段，使用 `CrossEntropyLoss` 作为损失函数。

- \*\*进度条\*\*：显示测试进度，共处理了 860 个批次的数据，处理速度约为 197.66 批次/秒，总耗时 4.4 秒。

- \*\*测试结果\*\*：测试集上的损失为 0.3881，准确率为 79.35%。

### 4. 统计信息保存与分析

```

Stats saved at stats-results/stats\_DistillBertEmotion\_2025-05-07\_22-03-00\_test\_crossentropy.csv

lowest\_stats\_df

original\_index text true\_label pred\_label prob loss

513 715 2228 5386 <UNK> <UNK> 3693 18132 3693 18132 49... 0 0 0.053552 0.055039

453 636 2228 3693 18132 3693 18132 7486 4887 4281 4000 0 0 0.054818 0.056377

233 337 2228 2906 <UNK> 6714 <UNK> 25811 5312 4953 534... 0 0 0.057254 0.058958

37 53 2228 24716 24955 5843 12581 2394 4512 5843 470... 0 0 0.059632 0.061484

354 506 2228 3693 18132 22852 4419 18155 3569 22852 64... 0 0 0.060784 0.062710

highest\_stats\_df

original\_index text true\_label pred\_label prob loss

425 599 2228 6181 <UNK> 4281 25022 <UNK> 5358 16987 <U... 0 1 0.902435 2.327241

502 700 2228 5091 5491 5858 5053 4281 5627 3810 3782 2... 0 1 0.886540 2.176303

514 716 2228 3693 18132 5843 26831 4512 3560 18155 151... 0 1 0.880616 2.125407

42 58 2228 5341 4953 19527 9694 3533 <UNK> 5671 4000 0 1 0.874697 2.077023

482 674 2228 3603 2827 5247 <UNK> <UNK> 19006 5247 946... 0 1 0.869784 2.038563

```

- \*\*统计信息保存\*\*：程序将测试结果保存到 `stats-results/stats\_DistillBertEmotion\_2025-05-07\_22-03-00\_test\_crossentropy.csv` 文件中。

- \*\*最低损失样本\*\*：`lowest\_stats\_df` 显示了损失最低的 5 个样本，这些样本的预测标签与真实标签一致，且预测概率较低，说明模型对这些样本的分类比较有把握。

- \*\*最高损失样本\*\*：`highest\_stats\_df` 显示了损失最高的 5 个样本，这些样本的预测标签与真实标签不一致，且预测概率较高，说明模型在这些样本上出现了错误分类。

### 总结

总体来说，程序成功完成了对 `DistillBertEmotion` 模型的测试，并保存了相关统计信息。测试结果显示，模型在测试集上的准确率为 79.35%，损失为 0.3881。通过分析损失最低和最高的样本，可以进一步了解模型的性能和可能存在的问题。