input for adv: mean of last 2 hidden layer

training 1 epoch...

/opt/conda/lib/python3.7/site-packages/ipykernel\_launcher.py:41: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires\_grad\_(True), rather than torch.tensor(sourceTensor).

93.6142725944519 seconds used......

1 training finished! average train loss: 0.6493212400298369

total clf loss: 151.95219260454178 total adv loss: 78.1390012614429

evaluating...

Confusion matrix, without normalization

[[ 229 44 21]

[ 733 2970 118]

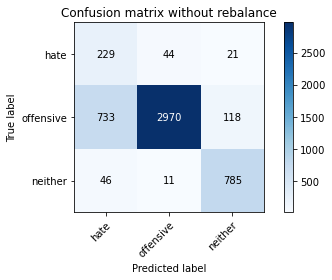
[ 46 11 785]]

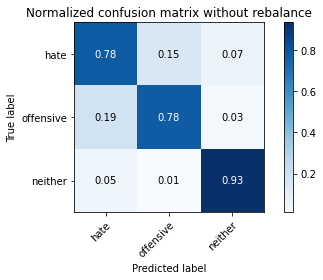
Normalized confusion matrix

[[0.77891156 0.14965986 0.07142857]

[0.1918346 0.77728343 0.03088197]

[0.05463183 0.01306413 0.93230404]]





precision recall f1-score support

hate 0.23 0.78 0.35 294

offensive 0.98 0.78 0.87 3821

neither 0.85 0.93 0.89 842

accuracy 0.80 4957

macro avg 0.69 0.83 0.70 4957

weighted avg 0.91 0.80 0.84 4957

average eval\_loss: 0.5003165734813653, accuracy: 0.8037119225337906

training 2 epoch...

/opt/conda/lib/python3.7/site-packages/ipykernel\_launcher.py:41: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires\_grad\_(True), rather than torch.tensor(sourceTensor).

93.71407771110535 seconds used......

2 training finished! average train loss: 0.37074101842090224

total clf loss: 86.39953845366836 total adv loss: 37.41172517463565

evaluating...

Confusion matrix, without normalization

[[ 244 41 9]

[ 864 2892 65]

[ 82 40 720]]

Normalized confusion matrix

[[0.82993197 0.13945578 0.03061224]

[0.22611882 0.75686993 0.01701125]

[0.09738717 0.04750594 0.85510689]]





precision recall f1-score support

hate 0.21 0.83 0.33 294

offensive 0.97 0.76 0.85 3821

neither 0.91 0.86 0.88 842

accuracy 0.78 4957

macro avg 0.69 0.81 0.69 4957

weighted avg 0.92 0.78 0.83 4957

average eval\_loss: 0.5010080802679242, accuracy: 0.7778898527335082

training 3 epoch...

/opt/conda/lib/python3.7/site-packages/ipykernel\_launcher.py:41: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires\_grad\_(True), rather than torch.tensor(sourceTensor).

93.85006666183472 seconds used......

3 training finished! average train loss: 0.26593877419241163

total clf loss: 61.82860862836242 total adv loss: 23.891364209819585

evaluating...

Confusion matrix, without normalization

[[ 175 91 28]

[ 342 3350 129]

[ 21 31 790]]

Normalized confusion matrix

[[0.5952381 0.30952381 0.0952381 ]

[0.08950537 0.87673384 0.0337608 ]

[0.02494062 0.0368171 0.93824228]]





precision recall f1-score support

hate 0.33 0.60 0.42 294

offensive 0.96 0.88 0.92 3821

neither 0.83 0.94 0.88 842

accuracy 0.87 4957

macro avg 0.71 0.80 0.74 4957

weighted avg 0.90 0.87 0.88 4957

average eval\_loss: 0.353479222402967, accuracy: 0.8704861811579584

training 4 epoch...

/opt/conda/lib/python3.7/site-packages/ipykernel\_launcher.py:41: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires\_grad\_(True), rather than torch.tensor(sourceTensor).

93.72024416923523 seconds used......

4 training finished! average train loss: 0.17184629987745562

total clf loss: 39.98564728721976 total adv loss: 16.09382032463327

evaluating...

Confusion matrix, without normalization

[[ 206 69 19]

[ 492 3219 110]

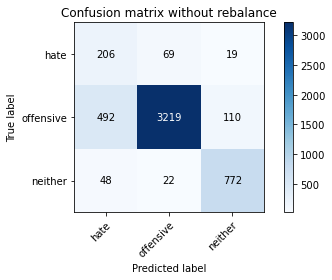
[ 48 22 772]]

Normalized confusion matrix

[[0.70068027 0.23469388 0.06462585]

[0.1287621 0.84244962 0.02878828]

[0.05700713 0.02612827 0.91686461]]





precision recall f1-score support

hate 0.28 0.70 0.40 294

offensive 0.97 0.84 0.90 3821

neither 0.86 0.92 0.89 842

accuracy 0.85 4957

macro avg 0.70 0.82 0.73 4957

weighted avg 0.91 0.85 0.87 4957

average eval\_loss: 0.47372379974149775, accuracy: 0.846681460560823

training 5 epoch...

/opt/conda/lib/python3.7/site-packages/ipykernel\_launcher.py:41: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires\_grad\_(True), rather than torch.tensor(sourceTensor).

93.71526217460632 seconds used......

5 training finished! average train loss: 0.10076096155599021

total clf loss: 23.28489847597666 total adv loss: 6.227985087083653

evaluating...

Confusion matrix, without normalization

[[ 235 41 18]

[ 878 2833 110]

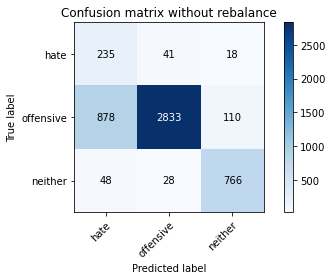
[ 48 28 766]]

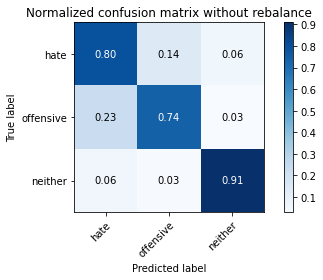
Normalized confusion matrix

[[0.79931973 0.13945578 0.06122449]

[0.22978278 0.74142895 0.02878828]

[0.05700713 0.03325416 0.90973872]]





precision recall f1-score support

hate 0.20 0.80 0.32 294

offensive 0.98 0.74 0.84 3821

neither 0.86 0.91 0.88 842

accuracy 0.77 4957

macro avg 0.68 0.82 0.68 4957

weighted avg 0.91 0.77 0.82 4957

average eval\_loss: 0.885088728994041, accuracy: 0.7734516844865846

training 6 epoch...

/opt/conda/lib/python3.7/site-packages/ipykernel\_launcher.py:41: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires\_grad\_(True), rather than torch.tensor(sourceTensor).

93.76580142974854 seconds used......

6 training finished! average train loss: 0.06268937200340524

total clf loss: 14.58990171784535 total adv loss: 5.9344995084102266

evaluating...

Confusion matrix, without normalization

[[ 258 29 7]

[1255 2525 41]

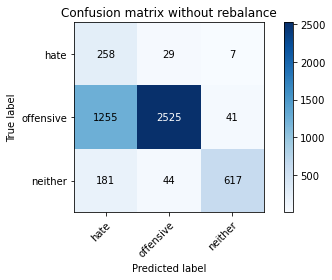
[ 181 44 617]]

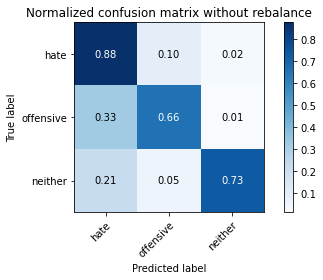
Normalized confusion matrix

[[0.87755102 0.09863946 0.02380952]

[0.32844805 0.66082177 0.01073018]

[0.21496437 0.05225653 0.7327791 ]]





precision recall f1-score support

hate 0.15 0.88 0.26 294

offensive 0.97 0.66 0.79 3821

neither 0.93 0.73 0.82 842

accuracy 0.69 4957

macro avg 0.68 0.76 0.62 4957

weighted avg 0.92 0.69 0.76 4957

average eval\_loss: 1.283096438127477, accuracy: 0.685898729070002

training 7 epoch...

/opt/conda/lib/python3.7/site-packages/ipykernel\_launcher.py:41: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires\_grad\_(True), rather than torch.tensor(sourceTensor).

93.74852275848389 seconds used......

7 training finished! average train loss: 0.056449868284848013

total clf loss: 13.253307568375021 total adv loss: 7.654751566937193

evaluating...

Confusion matrix, without normalization

[[ 207 71 16]

[ 633 3113 75]

[ 59 65 718]]

Normalized confusion matrix

[[0.70408163 0.2414966 0.05442177]

[0.16566344 0.81470819 0.01962837]

[0.07007126 0.07719715 0.85273159]]





precision recall f1-score support

hate 0.23 0.70 0.35 294

offensive 0.96 0.81 0.88 3821

neither 0.89 0.85 0.87 842

accuracy 0.81 4957

macro avg 0.69 0.79 0.70 4957

weighted avg 0.90 0.81 0.85 4957

average eval\_loss: 0.730730191350062, accuracy: 0.8146056082307848

training 8 epoch...

/opt/conda/lib/python3.7/site-packages/ipykernel\_launcher.py:41: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires\_grad\_(True), rather than torch.tensor(sourceTensor).

93.68621730804443 seconds used......

8 training finished! average train loss: 0.03633394998675948

total clf loss: 8.429256473085843 total adv loss: 2.902318786349497

evaluating...

Confusion matrix, without normalization

[[ 214 67 13]

[ 661 3079 81]

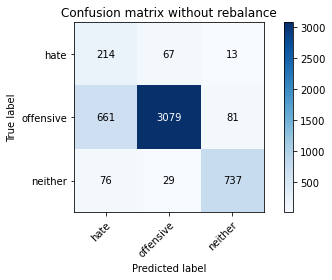
[ 76 29 737]]

Normalized confusion matrix

[[0.72789116 0.22789116 0.04421769]

[0.17299136 0.80581 0.02119864]

[0.09026128 0.03444181 0.87529691]]





precision recall f1-score support

hate 0.23 0.73 0.34 294

offensive 0.97 0.81 0.88 3821

neither 0.89 0.88 0.88 842

accuracy 0.81 4957

macro avg 0.69 0.80 0.70 4957

weighted avg 0.91 0.81 0.85 4957

average eval\_loss: 0.8490169399065877, accuracy: 0.8129917288682671

training 9 epoch...

/opt/conda/lib/python3.7/site-packages/ipykernel\_launcher.py:41: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires\_grad\_(True), rather than torch.tensor(sourceTensor).

93.62870693206787 seconds used......

9 training finished! average train loss: 0.023698122518365842

total clf loss: 5.542092255665921 total adv loss: 2.77840668745921

evaluating...

Confusion matrix, without normalization

[[ 221 59 14]

[ 743 2968 110]

[ 60 23 759]]

Normalized confusion matrix

[[0.75170068 0.20068027 0.04761905]

[0.19445171 0.77676001 0.02878828]

[0.07125891 0.02731591 0.90142518]]





precision recall f1-score support

hate 0.22 0.75 0.34 294

offensive 0.97 0.78 0.86 3821

neither 0.86 0.90 0.88 842

accuracy 0.80 4957

macro avg 0.68 0.81 0.69 4957

weighted avg 0.91 0.80 0.84 4957

average eval\_loss: 1.0442306735805023, accuracy: 0.7964494654024612

training 10 epoch...

/opt/conda/lib/python3.7/site-packages/ipykernel\_launcher.py:41: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires\_grad\_(True), rather than torch.tensor(sourceTensor).

93.5533378124237 seconds used......

10 training finished! average train loss: 0.011594636983206504

total clf loss: 2.681903860589955 total adv loss: 0.7665317420178326

evaluating...

Confusion matrix, without normalization

[[ 222 58 14]

[ 800 2925 96]

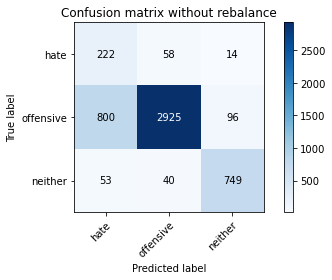
[ 53 40 749]]

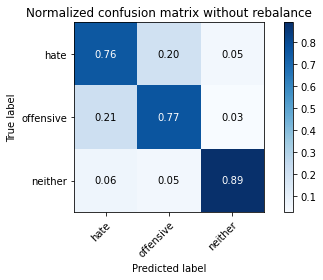
Normalized confusion matrix

[[0.75510204 0.19727891 0.04761905]

[0.20936928 0.76550641 0.02512431]

[0.06294537 0.04750594 0.88954869]]





precision recall f1-score support

hate 0.21 0.76 0.32 294

offensive 0.97 0.77 0.85 3821

neither 0.87 0.89 0.88 842

accuracy 0.79 4957

macro avg 0.68 0.80 0.69 4957

weighted avg 0.91 0.79 0.83 4957

average eval\_loss: 1.0724447918108846, accuracy: 0.7859592495460964