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).

196.2010953426361 seconds used......

1 training finished! average train loss: 0.6836621119526395

total clf loss: 159.8525091856718 total adv loss: 79.5509496703744

evaluating...

Confusion matrix, without normalization

[[ 205 57 32]

[ 610 3062 149]

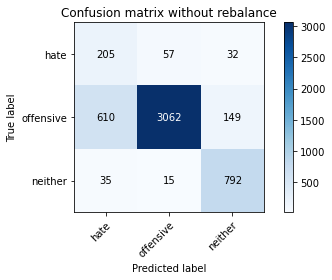
[ 35 15 792]]

Normalized confusion matrix

[[0.69727891 0.19387755 0.10884354]

[0.15964407 0.8013609 0.03899503]

[0.0415677 0.01781473 0.94061758]]





precision recall f1-score support

hate 0.24 0.70 0.36 294

offensive 0.98 0.80 0.88 3821

neither 0.81 0.94 0.87 842

accuracy 0.82 4957

macro avg 0.68 0.81 0.70 4957

weighted avg 0.91 0.82 0.85 4957

average eval\_loss: 0.42108470192488584, accuracy: 0.8188420415573936

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).

195.5170602798462 seconds used......

2 training finished! average train loss: 0.3968608108463517

total clf loss: 92.29646626114845 total adv loss: 36.244024742394686

evaluating...

Confusion matrix, without normalization

[[ 238 44 12]

[ 811 2962 48]

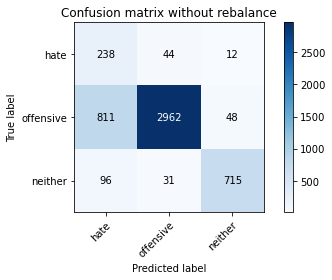
[ 96 31 715]]

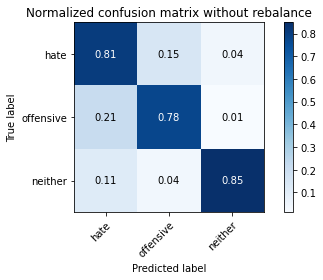
Normalized confusion matrix

[[0.80952381 0.14965986 0.04081633]

[0.2122481 0.77518974 0.01256216]

[0.11401425 0.0368171 0.84916865]]





precision recall f1-score support

hate 0.21 0.81 0.33 294

offensive 0.98 0.78 0.86 3821

neither 0.92 0.85 0.88 842

accuracy 0.79 4957

macro avg 0.70 0.81 0.69 4957

weighted avg 0.92 0.79 0.84 4957

average eval\_loss: 0.45861809122238273, accuracy: 0.7897922130320758

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).

195.5370490550995 seconds used......

3 training finished! average train loss: 0.28847342428931017

total clf loss: 66.93070027604699 total adv loss: 23.1751945578726

evaluating...

Confusion matrix, without normalization

[[ 216 48 30]

[ 686 3016 119]

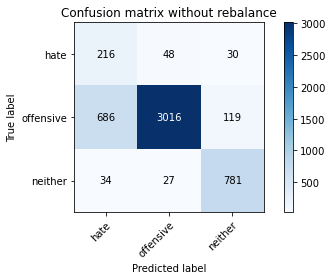
[ 34 27 781]]

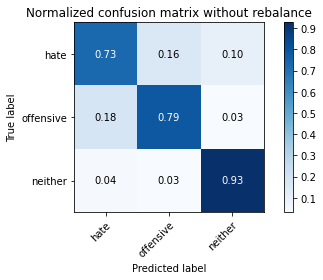
Normalized confusion matrix

[[0.73469388 0.16326531 0.10204082]

[0.17953415 0.78932217 0.03114368]

[0.04038005 0.03206651 0.92755344]]





precision recall f1-score support

hate 0.23 0.73 0.35 294

offensive 0.98 0.79 0.87 3821

neither 0.84 0.93 0.88 842

accuracy 0.81 4957

macro avg 0.68 0.82 0.70 4957

weighted avg 0.91 0.81 0.84 4957

average eval\_loss: 0.476628324005674, accuracy: 0.809562235222917

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).

195.57582759857178 seconds used......

4 training finished! average train loss: 0.18971252022311091

total clf loss: 46.782881919294596 total adv loss: 70.56854622997344

evaluating...

Confusion matrix, without normalization

[[ 217 56 21]

[ 653 3089 79]

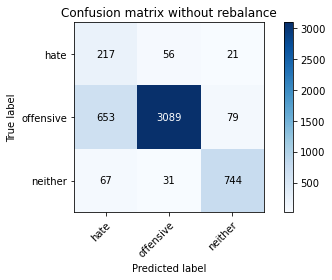
[ 67 31 744]]

Normalized confusion matrix

[[0.73809524 0.19047619 0.07142857]

[0.17089767 0.80842711 0.02067522]

[0.07957245 0.0368171 0.88361045]]





precision recall f1-score support

hate 0.23 0.74 0.35 294

offensive 0.97 0.81 0.88 3821

neither 0.88 0.88 0.88 842

accuracy 0.82 4957

macro avg 0.70 0.81 0.71 4957

weighted avg 0.91 0.82 0.85 4957

average eval\_loss: 0.48092803539387496, accuracy: 0.8170264272745612

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).

195.66245675086975 seconds used......

5 training finished! average train loss: 0.09860484846949316

total clf loss: 32.00013375189155 total adv loss: 190.36456145346165

evaluating...

Confusion matrix, without normalization

[[ 237 43 14]

[ 796 2947 78]

[ 87 24 731]]

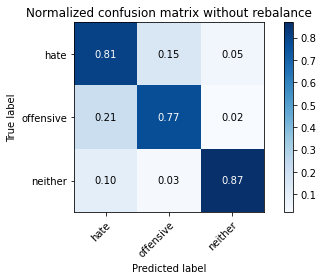
Normalized confusion matrix

[[0.80612245 0.1462585 0.04761905]

[0.20832243 0.77126407 0.0204135 ]

[0.10332542 0.02850356 0.86817102]]





precision recall f1-score support

hate 0.21 0.81 0.34 294

offensive 0.98 0.77 0.86 3821

neither 0.89 0.87 0.88 842

accuracy 0.79 4957

macro avg 0.69 0.82 0.69 4957

weighted avg 0.92 0.79 0.83 4957

average eval\_loss: 0.6155168411105286, accuracy: 0.7897922130320758

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).

195.6151032447815 seconds used......

6 training finished! average train loss: 0.049402133880281134

total clf loss: 18.712473794352263 total adv loss: 148.97574293613434

evaluating...

Confusion matrix, without normalization

[[ 221 55 18]

[ 742 2982 97]

[ 63 24 755]]

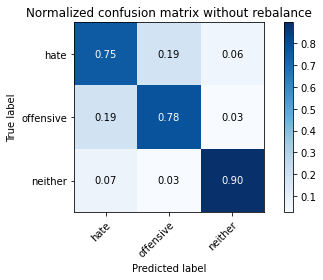
Normalized confusion matrix

[[0.75170068 0.18707483 0.06122449]

[0.19419 0.78042397 0.02538602]

[0.07482185 0.02850356 0.89667458]]





precision recall f1-score support

hate 0.22 0.75 0.33 294

offensive 0.97 0.78 0.87 3821

neither 0.87 0.90 0.88 842

accuracy 0.80 4957

macro avg 0.69 0.81 0.69 4957

weighted avg 0.91 0.80 0.84 4957

average eval\_loss: 0.7313432072836545, accuracy: 0.7984668146056082

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).

195.65472769737244 seconds used......

7 training finished! average train loss: 0.033020553603899064

total clf loss: 14.360886878101155 total adv loss: 136.6440103650093

evaluating...

Confusion matrix, without normalization

[[ 205 74 15]

[ 587 3161 73]

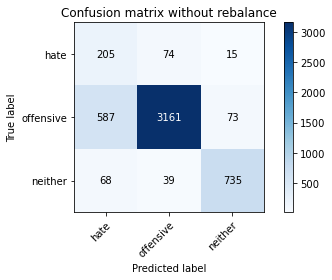
[ 68 39 735]]

Normalized confusion matrix

[[0.69727891 0.25170068 0.05102041]

[0.15362471 0.82727035 0.01910495]

[0.0807601 0.04631829 0.87292162]]





precision recall f1-score support

hate 0.24 0.70 0.36 294

offensive 0.97 0.83 0.89 3821

neither 0.89 0.87 0.88 842

accuracy 0.83 4957

macro avg 0.70 0.80 0.71 4957

weighted avg 0.91 0.83 0.86 4957

average eval\_loss: 0.6600643356251207, accuracy: 0.8273149082106113

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).

195.60614228248596 seconds used......

8 training finished! average train loss: 0.00603436491362293

total clf loss: 8.642049874993972 total adv loss: 145.3242917060852

evaluating...

Confusion matrix, without normalization

[[ 214 65 15]

[ 623 3126 72]

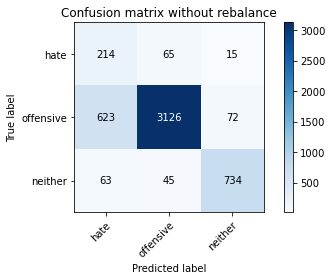
[ 63 45 734]]

Normalized confusion matrix

[[0.72789116 0.22108844 0.05102041]

[0.16304632 0.81811044 0.01884323]

[0.07482185 0.05344418 0.87173397]]





precision recall f1-score support

hate 0.24 0.73 0.36 294

offensive 0.97 0.82 0.89 3821

neither 0.89 0.87 0.88 842

accuracy 0.82 4957

macro avg 0.70 0.81 0.71 4957

weighted avg 0.91 0.82 0.85 4957

average eval\_loss: 0.7423838491856157, accuracy: 0.8218680653621142

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).

195.45995116233826 seconds used......

9 training finished! average train loss: -0.01367000915649298

total clf loss: 6.413415735354647 total adv loss: 190.6035534441471

evaluating...

Confusion matrix, without normalization

[[ 228 50 16]

[ 963 2775 83]

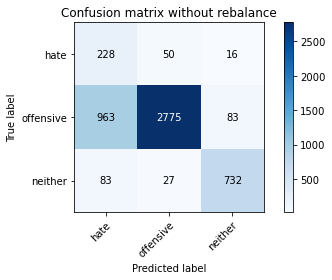
[ 83 27 732]]

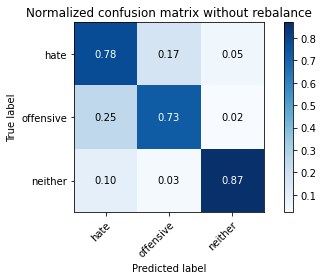
Normalized confusion matrix

[[0.7755102 0.17006803 0.05442177]

[0.25202826 0.72624967 0.02172206]

[0.09857482 0.03206651 0.86935867]]





precision recall f1-score support

hate 0.18 0.78 0.29 294

offensive 0.97 0.73 0.83 3821

neither 0.88 0.87 0.88 842

accuracy 0.75 4957

macro avg 0.68 0.79 0.67 4957

weighted avg 0.91 0.75 0.81 4957

average eval\_loss: 1.1995051435906767, accuracy: 0.7534799273754287