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).

84.85610342025757 seconds used......

1 training finished! average train loss: 0.644065070439849

total clf loss: 151.77194745838642 total adv loss: 98.50223285704851

evaluating...

Confusion matrix, without normalization

[[ 187 77 30]

[ 383 3298 140]

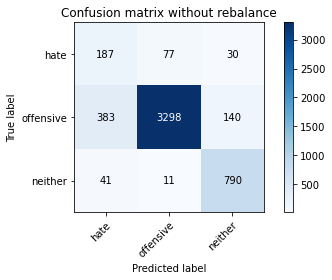
[ 41 11 790]]

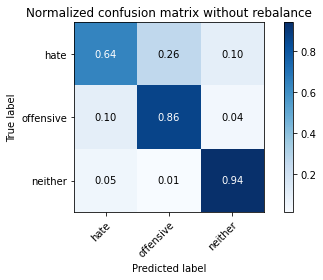
Normalized confusion matrix

[[0.63605442 0.26190476 0.10204082]

[0.10023554 0.86312484 0.03663962]

[0.04869359 0.01306413 0.93824228]]





precision recall f1-score support

hate 0.31 0.64 0.41 294

offensive 0.97 0.86 0.92 3821

neither 0.82 0.94 0.88 842

accuracy 0.86 4957

macro avg 0.70 0.81 0.74 4957

weighted avg 0.91 0.86 0.88 4957

average eval\_loss: 0.39428716497298816, accuracy: 0.8624167843453702

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).

84.41032075881958 seconds used......

2 training finished! average train loss: 0.37399256206526044

total clf loss: 88.67185612767935 total adv loss: 68.03104068338871

evaluating...

Confusion matrix, without normalization

[[ 243 32 19]

[ 905 2801 115]

[ 54 14 774]]

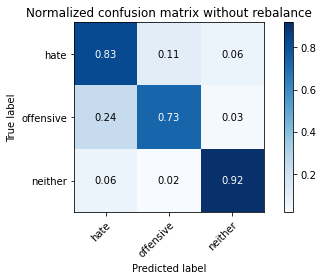
Normalized confusion matrix

[[0.82653061 0.10884354 0.06462585]

[0.23684899 0.73305417 0.03009683]

[0.06413302 0.01662708 0.9192399 ]]





precision recall f1-score support

hate 0.20 0.83 0.32 294

offensive 0.98 0.73 0.84 3821

neither 0.85 0.92 0.88 842

accuracy 0.77 4957

macro avg 0.68 0.83 0.68 4957

weighted avg 0.92 0.77 0.82 4957

average eval\_loss: 0.5368443120642371, accuracy: 0.7702239257615493

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).

84.42455005645752 seconds used......

3 training finished! average train loss: 0.24712817662542588

total clf loss: 58.897839250043035 total adv loss: 51.05229889601469

evaluating...

Confusion matrix, without normalization

[[ 234 42 18]

[ 845 2868 108]

[ 46 15 781]]

Normalized confusion matrix

[[0.79591837 0.14285714 0.06122449]

[0.2211463 0.75058885 0.02826485]

[0.05463183 0.01781473 0.92755344]]





precision recall f1-score support

hate 0.21 0.80 0.33 294

offensive 0.98 0.75 0.85 3821

neither 0.86 0.93 0.89 842

accuracy 0.78 4957

macro avg 0.68 0.82 0.69 4957

weighted avg 0.91 0.78 0.83 4957

average eval\_loss: 0.5162820197484285, accuracy: 0.7833366955820052

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).

84.37816381454468 seconds used......

4 training finished! average train loss: 0.15335345915273615

total clf loss: 36.777196186594665 total adv loss: 36.25215090112761

evaluating...

Confusion matrix, without normalization

[[ 208 63 23]

[ 572 3127 122]

[ 41 19 782]]

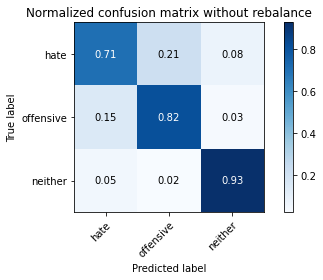
Normalized confusion matrix

[[0.70748299 0.21428571 0.07823129]

[0.14969903 0.81837215 0.03192881]

[0.04869359 0.02256532 0.92874109]]





precision recall f1-score support

hate 0.25 0.71 0.37 294

offensive 0.97 0.82 0.89 3821

neither 0.84 0.93 0.88 842

accuracy 0.83 4957

macro avg 0.69 0.82 0.72 4957

weighted avg 0.91 0.83 0.86 4957

average eval\_loss: 0.4640865475530216, accuracy: 0.8305426669356466

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).

84.2910647392273 seconds used......

5 training finished! average train loss: 0.08443962754436622

total clf loss: 20.30584949022159 total adv loss: 21.07229024625849

evaluating...

Confusion matrix, without normalization

[[ 218 61 15]

[ 618 3119 84]

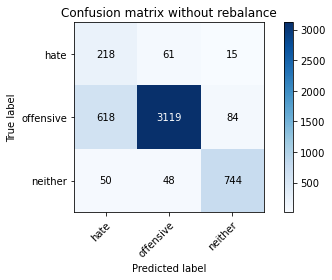
[ 50 48 744]]

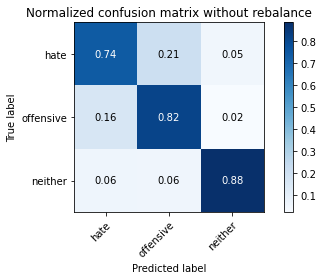
Normalized confusion matrix

[[0.7414966 0.20748299 0.05102041]

[0.16173776 0.81627846 0.02198377]

[0.05938242 0.05700713 0.88361045]]





precision recall f1-score support

hate 0.25 0.74 0.37 294

offensive 0.97 0.82 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: 0.5798149392954173, accuracy: 0.8232802098043172

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).

84.35286498069763 seconds used......

6 training finished! average train loss: 0.07220325616633586

total clf loss: 17.478619751054794 total adv loss: 20.32554769865237

evaluating...

Confusion matrix, without normalization

[[ 266 20 8]

[1631 2111 79]

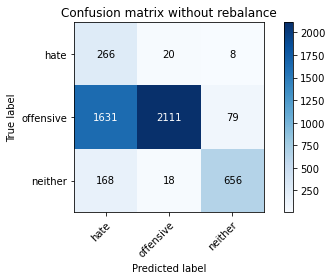
[ 168 18 656]]

Normalized confusion matrix

[[0.9047619 0.06802721 0.02721088]

[0.42685161 0.55247317 0.02067522]

[0.19952494 0.02137767 0.77909739]]





precision recall f1-score support

hate 0.13 0.90 0.23 294

offensive 0.98 0.55 0.71 3821

neither 0.88 0.78 0.83 842

accuracy 0.61 4957

macro avg 0.66 0.75 0.59 4957

weighted avg 0.91 0.61 0.70 4957

average eval\_loss: 1.7276588674083218, accuracy: 0.6118620133145047

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).

84.343425989151 seconds used......

7 training finished! average train loss: 0.054653772068201774

total clf loss: 13.896736353752203 total adv loss: 28.71352630748879

evaluating...

Confusion matrix, without normalization

[[ 189 76 29]

[ 482 3214 125]

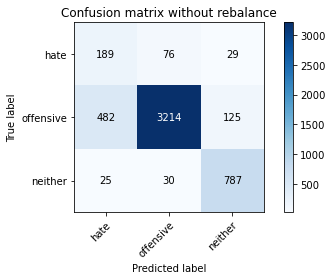
[ 25 30 787]]

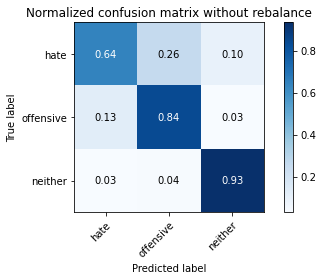
Normalized confusion matrix

[[0.64285714 0.2585034 0.09863946]

[0.12614499 0.84114106 0.03271395]

[0.02969121 0.03562945 0.93467933]]





precision recall f1-score support

hate 0.27 0.64 0.38 294

offensive 0.97 0.84 0.90 3821

neither 0.84 0.93 0.88 842

accuracy 0.85 4957

macro avg 0.69 0.81 0.72 4957

weighted avg 0.90 0.85 0.87 4957

average eval\_loss: 0.5923988393028168, accuracy: 0.8452693161186201

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).

84.55962467193604 seconds used......

8 training finished! average train loss: 0.018868729765241017

total clf loss: 6.053368727094494 total adv loss: 35.0259663334582

evaluating...

Confusion matrix, without normalization

[[ 210 65 19]

[ 602 3090 129]

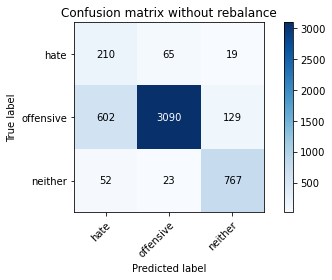
[ 52 23 767]]

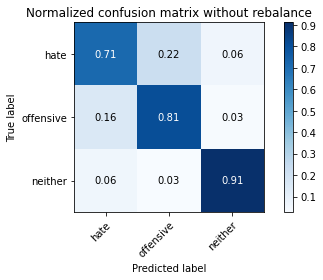
Normalized confusion matrix

[[0.71428571 0.22108844 0.06462585]

[0.15755038 0.80868882 0.0337608 ]

[0.06175772 0.02731591 0.91092637]]





precision recall f1-score support

hate 0.24 0.71 0.36 294

offensive 0.97 0.81 0.88 3821

neither 0.84 0.91 0.87 842

accuracy 0.82 4957

macro avg 0.68 0.81 0.71 4957

weighted avg 0.91 0.82 0.85 4957

average eval\_loss: 0.8208562688004462, accuracy: 0.8204559209199113

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).

84.49315690994263 seconds used......

9 training finished! average train loss: 0.018720673968537563

total clf loss: 7.386524164059665 total adv loss: 62.36420921329409

evaluating...

Confusion matrix, without normalization

[[ 230 45 19]

[ 837 2842 142]

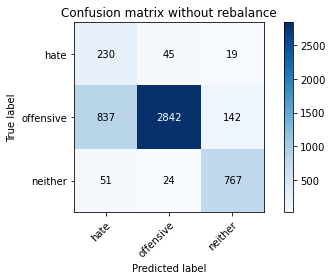
[ 51 24 767]]

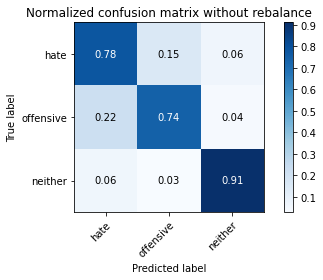
Normalized confusion matrix

[[0.78231293 0.15306122 0.06462585]

[0.2190526 0.74378435 0.03716305]

[0.06057007 0.02850356 0.91092637]]





precision recall f1-score support

hate 0.21 0.78 0.33 294

offensive 0.98 0.74 0.84 3821

neither 0.83 0.91 0.87 842

accuracy 0.77 4957

macro avg 0.67 0.81 0.68 4957

weighted avg 0.91 0.77 0.82 4957

average eval\_loss: 0.9337635917787431, accuracy: 0.7744603590881581

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).

84.366290807724 seconds used......

10 training finished! average train loss: 0.004496163126947195

total clf loss: 6.742684713797644 total adv loss: 114.35118918120861

evaluating...

Confusion matrix, without normalization

[[ 206 69 19]

[ 681 2999 141]

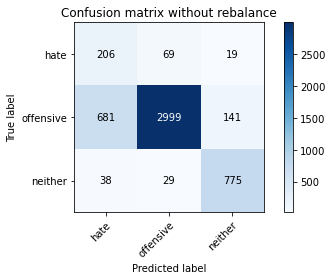
[ 38 29 775]]

Normalized confusion matrix

[[0.70068027 0.23469388 0.06462585]

[0.1782256 0.78487307 0.03690133]

[0.04513064 0.03444181 0.92042755]]





precision recall f1-score support

hate 0.22 0.70 0.34 294

offensive 0.97 0.78 0.87 3821

neither 0.83 0.92 0.87 842

accuracy 0.80 4957

macro avg 0.67 0.80 0.69 4957

weighted avg 0.90 0.80 0.84 4957

average eval\_loss: 0.9031055424534055, accuracy: 0.8029049828525318