pooled output as input for lstm

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).

85.59709882736206 seconds used......

1 training finished! average train loss: 0.6525440306255692

total clf loss: 152.7887743934989 total adv loss: 80.1747050974518

evaluating...

Confusion matrix, without normalization

[[ 217 59 18]

[ 635 3108 78]

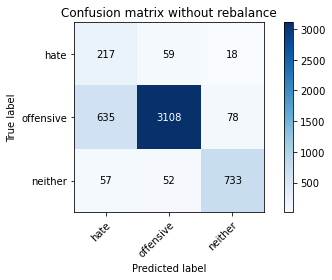
[ 57 52 733]]

Normalized confusion matrix

[[0.73809524 0.20068027 0.06122449]

[0.16618686 0.81339963 0.0204135 ]

[0.06769596 0.06175772 0.87054632]]





precision recall f1-score support

hate 0.24 0.74 0.36 294

offensive 0.97 0.81 0.88 3821

neither 0.88 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: 0.440373310487256, accuracy: 0.8186403066370789

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).

85.58630752563477 seconds used......

2 training finished! average train loss: 0.3798216667941265

total clf loss: 88.37429767847061 total adv loss: 35.499147882219404

evaluating...

Confusion matrix, without normalization

[[ 219 68 7]

[ 596 3181 44]

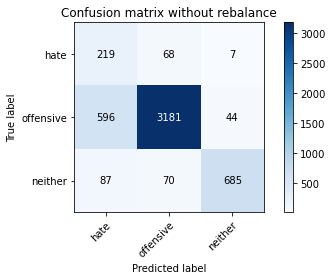
[ 87 70 685]]

Normalized confusion matrix

[[0.74489796 0.23129252 0.02380952]

[0.15598011 0.83250458 0.01151531]

[0.10332542 0.08313539 0.81353919]]





precision recall f1-score support

hate 0.24 0.74 0.37 294

offensive 0.96 0.83 0.89 3821

neither 0.93 0.81 0.87 842

accuracy 0.82 4957

macro avg 0.71 0.80 0.71 4957

weighted avg 0.91 0.82 0.86 4957

average eval\_loss: 0.4285187332010251, accuracy: 0.8240871494855759

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).

85.47897911071777 seconds used......

3 training finished! average train loss: 0.26308914287048474

total clf loss: 60.90562438219786 total adv loss: 18.42599654663354

evaluating...

Confusion matrix, without normalization

[[ 238 36 20]

[ 871 2842 108]

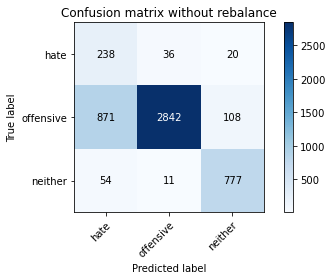
[ 54 11 777]]

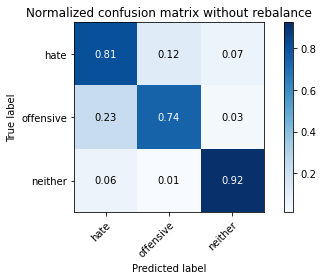
Normalized confusion matrix

[[0.80952381 0.12244898 0.06802721]

[0.2279508 0.74378435 0.02826485]

[0.06413302 0.01306413 0.92280285]]





precision recall f1-score support

hate 0.20 0.81 0.33 294

offensive 0.98 0.74 0.85 3821

neither 0.86 0.92 0.89 842

accuracy 0.78 4957

macro avg 0.68 0.83 0.69 4957

weighted avg 0.92 0.78 0.82 4957

average eval\_loss: 0.5360657233868661, accuracy: 0.7780915876538229

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).

85.56691193580627 seconds used......

4 training finished! average train loss: 0.17773049111118572

total clf loss: 41.12676828727126 total adv loss: 12.084325739531778

evaluating...

Confusion matrix, without normalization

[[ 243 34 17]

[1019 2656 146]

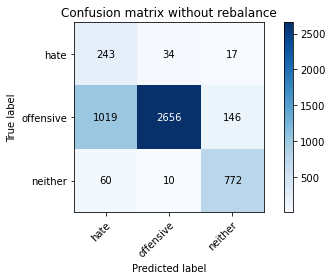
[ 60 10 772]]

Normalized confusion matrix

[[0.82653061 0.11564626 0.05782313]

[0.26668411 0.69510599 0.03820989]

[0.07125891 0.01187648 0.91686461]]





precision recall f1-score support

hate 0.18 0.83 0.30 294

offensive 0.98 0.70 0.81 3821

neither 0.83 0.92 0.87 842

accuracy 0.74 4957

macro avg 0.66 0.81 0.66 4957

weighted avg 0.91 0.74 0.79 4957

average eval\_loss: 0.6963578462938348, accuracy: 0.7405688924752875

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).

85.40545773506165 seconds used......

5 training finished! average train loss: 0.09727301695710865

total clf loss: 22.462868462782353 total adv loss: 5.692408865550533

evaluating...

Confusion matrix, without normalization

[[ 234 50 10]

[ 854 2910 57]

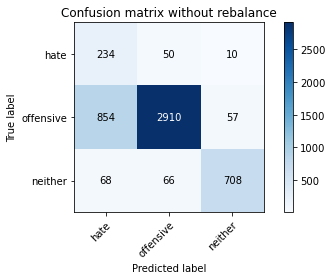
[ 68 66 708]]

Normalized confusion matrix

[[0.79591837 0.17006803 0.03401361]

[0.2235017 0.76158074 0.01491756]

[0.0807601 0.0783848 0.84085511]]





precision recall f1-score support

hate 0.20 0.80 0.32 294

offensive 0.96 0.76 0.85 3821

neither 0.91 0.84 0.88 842

accuracy 0.78 4957

macro avg 0.69 0.80 0.68 4957

weighted avg 0.91 0.78 0.82 4957

average eval\_loss: 0.6413199041528329, accuracy: 0.7770829130522493

**last hidden state as input for adv**

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).

88.95075368881226 seconds used......

1 training finished! average train loss: 0.6875088047562984

total clf loss: 160.8254704028368 total adv loss: 81.46926457807422

evaluating...

Confusion matrix, without normalization

[[ 240 22 32]

[1645 2029 147]

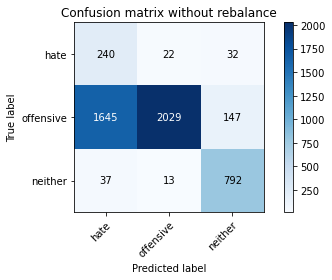
[ 37 13 792]]

Normalized confusion matrix

[[0.81632653 0.07482993 0.10884354]

[0.43051557 0.53101282 0.0384716 ]

[0.04394299 0.01543943 0.94061758]]





precision recall f1-score support

hate 0.12 0.82 0.22 294

offensive 0.98 0.53 0.69 3821

neither 0.82 0.94 0.87 842

accuracy 0.62 4957

macro avg 0.64 0.76 0.59 4957

weighted avg 0.90 0.62 0.69 4957

average eval\_loss: 0.8072024199807752, accuracy: 0.6175105910833165

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).

89.01979756355286 seconds used......

2 training finished! average train loss: 0.41317387879417655

total clf loss: 96.46099201589823 total adv loss: 45.14694920927286

evaluating...

Confusion matrix, without normalization

[[ 211 50 33]

[ 663 2981 177]

[ 17 17 808]]

Normalized confusion matrix

[[0.71768707 0.17006803 0.1122449 ]

[0.17351479 0.78016226 0.04632295]

[0.02019002 0.02019002 0.95961995]]





precision recall f1-score support

hate 0.24 0.72 0.36 294

offensive 0.98 0.78 0.87 3821

neither 0.79 0.96 0.87 842

accuracy 0.81 4957

macro avg 0.67 0.82 0.70 4957

weighted avg 0.90 0.81 0.84 4957

average eval\_loss: 0.4329796951950677, accuracy: 0.8069396812588259

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).

89.21744298934937 seconds used......

3 training finished! average train loss: 0.29008629978552725

total clf loss: 67.65902996435761 total adv loss: 30.387072215788066

evaluating...

Confusion matrix, without normalization

[[ 251 35 8]

[ 969 2785 67]

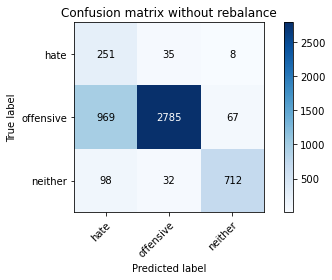
[ 98 32 712]]

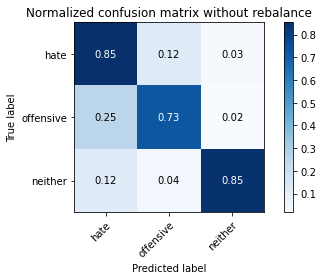
Normalized confusion matrix

[[0.8537415 0.11904762 0.02721088]

[0.25359853 0.72886679 0.01753468]

[0.11638955 0.03800475 0.8456057 ]]





precision recall f1-score support

hate 0.19 0.85 0.31 294

offensive 0.98 0.73 0.83 3821

neither 0.90 0.85 0.87 842

accuracy 0.76 4957

macro avg 0.69 0.81 0.67 4957

weighted avg 0.92 0.76 0.81 4957

average eval\_loss: 0.5464264921213745, accuracy: 0.7561024813395198

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).

88.85755968093872 seconds used......

4 training finished! average train loss: 0.20431596262936005

total clf loss: 47.57817796804011 total adv loss: 19.882767451694235

evaluating...

Confusion matrix, without normalization

[[ 239 40 15]

[ 944 2784 93]

[ 55 27 760]]

Normalized confusion matrix

[[0.81292517 0.13605442 0.05102041]

[0.24705574 0.72860508 0.02433918]

[0.06532067 0.03206651 0.90261283]]





precision recall f1-score support

hate 0.19 0.81 0.31 294

offensive 0.98 0.73 0.83 3821

neither 0.88 0.90 0.89 842

accuracy 0.76 4957

macro avg 0.68 0.81 0.68 4957

weighted avg 0.91 0.76 0.81 4957

average eval\_loss: 0.5705294514924648, accuracy: 0.7631632035505346

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).

88.71507430076599 seconds used......

5 training finished! average train loss: 0.1270385095624972

total clf loss: 29.4695625747554 total adv loss: 10.09564717579633

evaluating...

Confusion matrix, without normalization

[[ 220 50 24]

[ 852 2812 157]

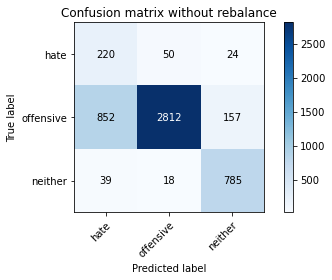
[ 39 18 785]]

Normalized confusion matrix

[[0.74829932 0.17006803 0.08163265]

[0.22297828 0.735933 0.04108872]

[0.04631829 0.02137767 0.93230404]]





precision recall f1-score support

hate 0.20 0.75 0.31 294

offensive 0.98 0.74 0.84 3821

neither 0.81 0.93 0.87 842

accuracy 0.77 4957

macro avg 0.66 0.81 0.67 4957

weighted avg 0.90 0.77 0.81 4957

average eval\_loss: 0.7016827737732462, accuracy: 0.7700221908412346

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).

88.90888142585754 seconds used......

6 training finished! average train loss: 0.07725176814200128

total clf loss: 17.961671247612685 total adv loss: 6.965362312155776

evaluating...

Confusion matrix, without normalization

[[ 231 45 18]

[ 893 2833 95]

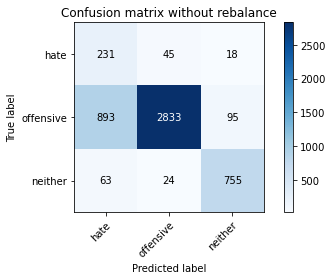
[ 63 24 755]]

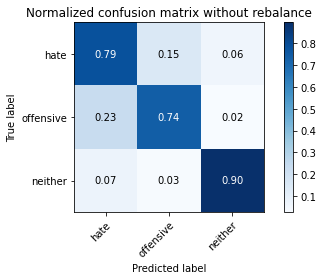
Normalized confusion matrix

[[0.78571429 0.15306122 0.06122449]

[0.23370845 0.74142895 0.0248626 ]

[0.07482185 0.02850356 0.89667458]]





precision recall f1-score support

hate 0.19 0.79 0.31 294

offensive 0.98 0.74 0.84 3821

neither 0.87 0.90 0.88 842

accuracy 0.77 4957

macro avg 0.68 0.81 0.68 4957

weighted avg 0.91 0.77 0.82 4957

average eval\_loss: 0.790122537049282, accuracy: 0.770425660681864

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).

88.83153939247131 seconds used......

7 training finished! average train loss: 0.04156333726642882

total clf loss: 9.660546993138269 total adv loss: 3.682120608224068

evaluating...

Confusion matrix, without normalization

[[ 211 59 24]

[ 751 2944 126]

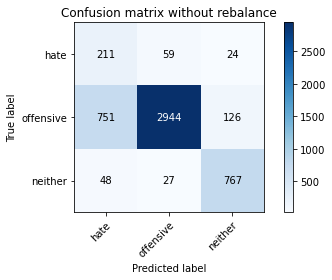
[ 48 27 767]]

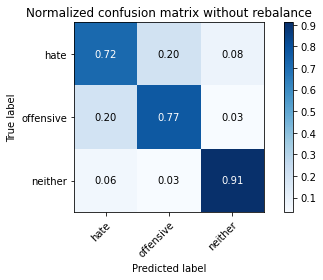
Normalized confusion matrix

[[0.71768707 0.20068027 0.08163265]

[0.19654541 0.77047893 0.03297566]

[0.05700713 0.03206651 0.91092637]]





precision recall f1-score support

hate 0.21 0.72 0.32 294

offensive 0.97 0.77 0.86 3821

neither 0.84 0.91 0.87 842

accuracy 0.79 4957

macro avg 0.67 0.80 0.69 4957

weighted avg 0.90 0.79 0.83 4957

average eval\_loss: 0.8376232022187444, accuracy: 0.7912043574742788

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).

88.7542233467102 seconds used......

8 training finished! average train loss: 0.039955700051219185

total clf loss: 9.288052751217037 total adv loss: 3.5630616648268187

evaluating...

Confusion matrix, without normalization

[[ 204 68 22]

[ 595 3115 111]

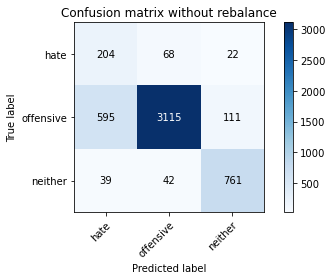
[ 39 42 761]]

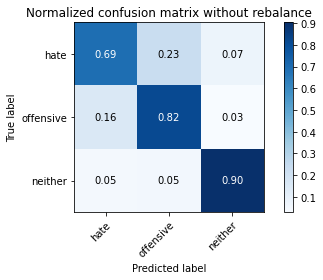
Normalized confusion matrix

[[0.69387755 0.23129252 0.07482993]

[0.1557184 0.81523161 0.02904999]

[0.04631829 0.04988124 0.90380048]]





precision recall f1-score support

hate 0.24 0.69 0.36 294

offensive 0.97 0.82 0.88 3821

neither 0.85 0.90 0.88 842

accuracy 0.82 4957

macro avg 0.69 0.80 0.71 4957

weighted avg 0.90 0.82 0.85 4957

average eval\_loss: 0.7652232390191839, accuracy: 0.8230784748840024

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).

88.61328458786011 seconds used......

9 training finished! average train loss: 0.03863505189707339

total clf loss: 9.126073059625924 total adv loss: 6.345625361951534

evaluating...

Confusion matrix, without normalization

[[ 223 51 20]

[ 729 2973 119]

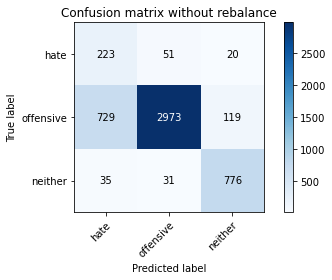
[ 35 31 776]]

Normalized confusion matrix

[[0.7585034 0.17346939 0.06802721]

[0.19078775 0.77806857 0.03114368]

[0.0415677 0.0368171 0.9216152 ]]





precision recall f1-score support

hate 0.23 0.76 0.35 294

offensive 0.97 0.78 0.86 3821

neither 0.85 0.92 0.88 842

accuracy 0.80 4957

macro avg 0.68 0.82 0.70 4957

weighted avg 0.91 0.80 0.84 4957

average eval\_loss: 0.9014070870300609, accuracy: 0.8012911034900141

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).

88.80662941932678 seconds used......

10 training finished! average train loss: 0.021487235767141868

total clf loss: 5.047757425694726 total adv loss: 2.973353778856108

evaluating...

Confusion matrix, without normalization

[[ 246 32 16]

[1060 2643 118]

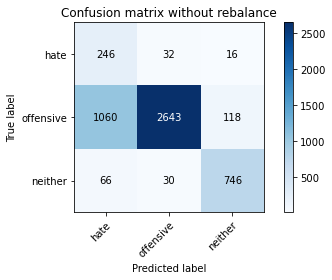
[ 66 30 746]]

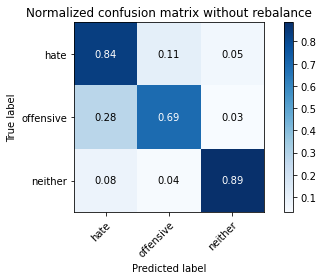
Normalized confusion matrix

[[0.83673469 0.10884354 0.05442177]

[0.27741429 0.69170374 0.03088197]

[0.0783848 0.03562945 0.88598575]]





precision recall f1-score support

hate 0.18 0.84 0.30 294

offensive 0.98 0.69 0.81 3821

neither 0.85 0.89 0.87 842

accuracy 0.73 4957

macro avg 0.67 0.80 0.66 4957

weighted avg 0.91 0.73 0.79 4957

average eval\_loss: 1.367088836533358, accuracy: 0.733306435343958