My local machine is an old i7 4790k, 24GB ram, and new-architecture medium-range RTX 3060 TI 8GB. The CuDNN backend was used with the higher level Tensorflow. The included code runs reasonable fast, though stability is not guaranteed for lower spec machines. All non TensorFlow code can be run on Collaboratory, though the TensorFlow code might not run due to Colab’s memory limitations.

LSTM weights as over 400mb to 1gb. They are not provided, as a deep learning capable machine can get a nearly as good fit in less then

Training with batch\_size=256 is only essential for 10 epochs, and is 15 to 25 secounds per epoch. That is faster than I can download from google drive. Let alone the marker’s work have having to match weights to dataset (and chance of error)

history = model.fit(X\_train, y\_train, validation\_split=0.05, epochs=10, batch\_size=256, verbose=1, callbacks=[lr, cp])

This original data is from: https://www.kaggle.com/stackoverflow/stacksample

preprocessed data can be obtained from: <https://drive.google.com/drive/folders/1N85lMKVfxQSIn7hfRzdzuXNo-fSUHGQ5?usp=sharing>

note the MLPClassifier at the very end might takes a very long time. Only say to save it is via pickle, which is a security issue to share. You may wish to skim. I left comments with outputs

on the full data set with **Tokenized**

all data is in my paper. I don’t have high decimal numbers saved.

on the full data set with **TfidfVectorizer**

min\_df=200 # 8908 dims

Clf: ExtraTreesClassifier

Accuracy score: 0.22334586682765653

Recall score: 0.27799837100386887

Precision score: 0.7935911481842874

F1 score: 0.3873825144566118

Clf: RandomForestClassifier

Accuracy score: 0.27327359674546653

Recall score: 0.35183539734547675

Precision score: 0.7965338057751677

F1 score: 0.4543767120735285

0.4543767120735285Train score: 0.9516728624535316

Test score: 0.27327359674546653

Time taken for RandomForestClassifier was 539.40

Clf: MLPClassifier

Accuracy score: 0.3619516178728798

Recall score: 0.5404240943336851

Precision score: 0.7520324734933328

F1 score: 0.6189265432694974

0.6189265432694974Train score: 0.4378400585176204

Test score: 0.3619516178728798

Time taken for MLPClassifier was 165.39

on the top 3 score data set with **tokenized**

Clf: ExtraTreesClassifier

Accuracy score: 6.245169751520308e-05

Recall score: 9.295726412094229e-05

Precision score: 0.15857315788576337

F1 score: 0.0001857227195545714

Train score: 1.0

Test score: 6.245169751520308e-05

Time taken for ExtraTreesClassifier was 31.04

Clf: RandomForestClassifier

Accuracy score: 0.0006010975885838297

Recall score: 0.0005675285598962793

Precision score: 0.23681389711605078

F1 score: 0.001129585405554267

Train score: 0.8762490632025981

Test score: 0.0006010975885838297

Time taken for RandomForestClassifier was 41.99

Clf: MLPClassifier

Accuracy score: 0.0

Recall score: 0.0008023679639912914

Precision score: 0.04850754334439879

F1 score: 0.0015413825206534702

0.0015413825206534702

Train score: 0.0

Test score: 0.0

Time taken for MLPClassifier was 54.98

on the top 3 score data set with **TfidfVectorizer**

Clf: ExtraTreesClassifier

Accuracy score: 0.2486494925839188

Recall score: 0.29594565578090537

Precision score: 0.7973873461949164

F1 score: 0.40604024121287086

Train score: 1.0

Test score: 0.2486494925839188

Time taken for ExtraTreesClassifier was 157.88

Clf: RandomForestClassifier

Accuracy score: 0.30379391100702574

Recall score: 0.37855511526674346

Precision score: 0.7756578961481374

F1 score: 0.47787457166061137

Train score: 0.9530906025074553

Test score: 0.30379391100702574

Time taken for RandomForestClassifier was 177.91

Clf: MLPClassifier

Accuracy score: 0.3590007806401249

Recall score: 0.5912861855589607

Precision score: 0.6850863021757897

F1 score: 0.6321836389630043

Train score: 0.899163140720386

Test score: 0.3590007806401249

Time taken for MLPClassifier was 1085.79

The below are at over 40 epochs and after training embeddings, decreasing both batch size and LR. My eval code is focused on **speed**, you should get nearly as good performance.

Regarding the LSTMon the **all scores** dataset "questions\_preprocessed.csv"

2963/2963 [==============================] - 85s 29ms/step - loss: 0.0199 - acc: 0.6533 - val\_loss: 0.0242 - val\_acc: 0.6135 - lr: 5.0000e-04

Accuracy score: 0.6504909819639279

Recall score: 0.7615028687766056

Precision score: 0.7574494222705919

F1 score: 0.7544918807953225

Regarding the LSTM on the **min 3 scores** dataset "questions\_preprocessed\_min3.csv"

1902/1902 [==============================] - 47s 25ms/step - loss: 0.0195 - acc: 0.6716 - val\_loss: 0.0313 - val\_acc: 0.5739 - lr: 1.5000e-04

Accuracy score: 0.6123965651834504

Recall score: 0.7221690839173125

Precision score: 0.7139613396235852

F1 score: 0.714711189087093