**Sentiment Analysis of Movie Reviews Using LSTM**

**NLP PROJECT REPORT: HXV180001**

1. **Successfully Installed the Environment and Executed the package : Yes**

Downloaded the glove model from the link and successfully incorporated it in project, sample execution of project given below.

**Output with default Config execution :**

Namespace(batch\_size=64, dpout\_fc=0.0, dpout\_model=0.0, enc\_lstm\_dim=128, encoder\_type='LSTMEncoder', fc\_dim=64, n\_classes=2, n\_enc\_layers=1, n\_epochs=20, nlipath='dataset/stsa/', nonlinear\_fc=0, optimizer='adam', outputdir='savedir/', outputmodelname='model.pickle', pool\_type='max', seed=1234, word\_emb\_dim=300, word\_emb\_path='dataset/GloVe/glove.840B.300d.txt')

\*\* TRAIN DATA : Found 6920 pairs of train sentences.

\*\* DEV DATA : Found 872 pairs of dev sentences.

\*\* TEST DATA : Found 1821 pairs of test sentences.

Found 16517(/17576) words with glove vectors

Vocab size : 16517

NLINet(

(encoder): LSTMEncoder(

(enc\_lstm): LSTM(300, 128)

)

(classifier): Sequential(

(0): Linear(in\_features=128, out\_features=64, bias=True)

(1): Linear(in\_features=64, out\_features=64, bias=True)

(2): Linear(in\_features=64, out\_features=2, bias=True)

)

)

VALIDATION : Epoch 1000000.0

finalgrep : accuracy valid : 84.0596

finalgrep : accuracy test : 84.6238

1. **Have you made modifications on the hyperparameters? : Yes** (Multiples)
2. **Increasing the epoch to 40**

**Result :**

**No impact as no changes has been made to model structure also the data is less so model will not perform better than that even if the epoch is increased.**

Namespace(batch\_size=64, dpout\_fc=0.0, dpout\_model=0.0, enc\_lstm\_dim=128, encoder\_type='LSTMEncoder', fc\_dim=64, n\_classes=2, n\_enc\_layers=1, n\_epochs=40, nlipath='dataset/stsa/', nonlinear\_fc=0, optimizer='adam', outputdir='savedir/', outputmodelname='model.pickle', pool\_type='max', seed=1234, word\_emb\_dim=300, word\_emb\_path='dataset/GloVe/glove.840B.300d.txt')

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 84.0596**

**finalgrep : accuracy test : 84.6238**

* **nothing changes as such in terms of accuracy.**

1. **Increasing batch size to 128 with epoch : 20**

**NLINet(**

**(encoder): LSTMEncoder(**

**(enc\_lstm): LSTM(300, 128)**

**)**

**(classifier): Sequential(**

**(0): Linear(in\_features=128, out\_features=64, bias=True)**

**(1): Linear(in\_features=64, out\_features=64, bias=True)**

**(2): Linear(in\_features=64, out\_features=2, bias=True)**

**))**

**Result :**

After 20 epochs the accuracy increases in test data from **84.6238** to **84.7337 , as the batch size increased so model will train faster on given data.**

**TEST : Epoch 21**

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 82.6835**

**finalgrep : accuracy test : 84.7337**

1. **Adding dropout in encoder, setting --dpout\_model = 1**

**Result** : Accuracy increases from previous 82% to 84.7%, adding dropout in classifier will make model neurons richer and comprehensive.

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 82.6835**

**finalgrep : accuracy test : 84.7337**

1. **Adding Dropout in classifier, --dpout\_fc = 1 and** –dpout\_model = 1, batch\_size =128

**Result**: No significant change

NLINet(

(encoder): LSTMEncoder(

(enc\_lstm): LSTM(300, 128, dropout=1.0)

)

(classifier): Sequential(

(0): Linear(in\_features=128, out\_features=64, bias=True)

(1): Linear(in\_features=64, out\_features=64, bias=True)

(2): Linear(in\_features=64, out\_features=2, bias=True)

)

)

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 82.6835**

**finalgrep : accuracy test : 84.7337**

1. **Adding non-linerality to be = 1**

**Result :** Accuracy will boost a bit as model will have more layers to train but again the training time will also increase.

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 82.6835**

**finalgrep : accuracy test : 84.7337**

1. **--optimizer", type=str, default="adamax"**

Changing optimizer with learning rate of 0.8, standard gradient desent optimizer

**Result:** As learning rate was 0.8 the model will learn fast and gradient steps will be higher towards optimum, no such impact on accuracy as training data doesn’t change also model architectural level remains same.

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 83.4862**

**finalgrep : accuracy test : 83.3608**

1. **Adding adam optimizer, --dpout model : 1 (Encoder Dropout) ,** parser.add\_argument(**"--optimizer"**, type=str, default=**"adam"**, help=**"adam or sgd,lr=0.1"**)**:**

**NLINet(**

**(encoder): LSTMEncoder(**

**(enc\_lstm): LSTM(300, 128, dropout=0.4)**

**)**

**(classifier): Sequential(**

**(0): Linear(in\_features=128, out\_features=64, bias=True)**

**(1): Linear(in\_features=64, out\_features=64, bias=True)**

**(2): Linear(in\_features=64, out\_features=2, bias=True)**

**)**

**)**

**Result :**

Changing to adam optimizer with batch-size = 128 and dropout in layers to 0.4 the model boost some accuracy and raises up to 84.7 from previous 82.6. The dropout helps to retain more and more neurons active in neural network.

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 82.6835**

**finalgrep : accuracy test : 84.7337**

**Increased accuracy from 84.6238** to **84.7337**

1. **Increasing encoder batch size to 256,Changing encoder input dimensions.**

parser.add\_argument(**"--enc\_lstm\_dim"**, type=int, default=256, help=**"encoder nhid dimension"**)

**Result:**

No significant impact on accuracy as batch size will just increasing training data per iteration and model will converge soon.

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 83.4862**

**finalgrep : accuracy test : 83.5255**

1. **Reducing encoder batch size to 64,Changing encoder input dimensions.**

parser.add\_argument(**"--enc\_lstm\_dim"**, type=int, default=64, help=**"encoder nhid dimension"**)

**Result:**

No significant impact on accuracy as batch size will just reduce training data per iteration and model will converge later and need more epochs to converge.

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 82.5688**

**finalgrep : accuracy test : 83.9099**

1. **Changing pool-type to mean :**

**Result :**

**No such impact on accuracy as the training data is not huge.**

parser.add\_argument(**"--pool\_type"**, type=str, default=**'mean'**, help=**"max or mean"**)

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 82.6835**

**finalgrep : accuracy test : 84.7337**

1. **Changing fc-dim to 128:**

parser.add\_argument(**"--fc\_dim"**, type=int, default=128, help=**"nhid of fc layers"**)

**Result :**

**No such impact on accuracy as it is not considered in model building.**

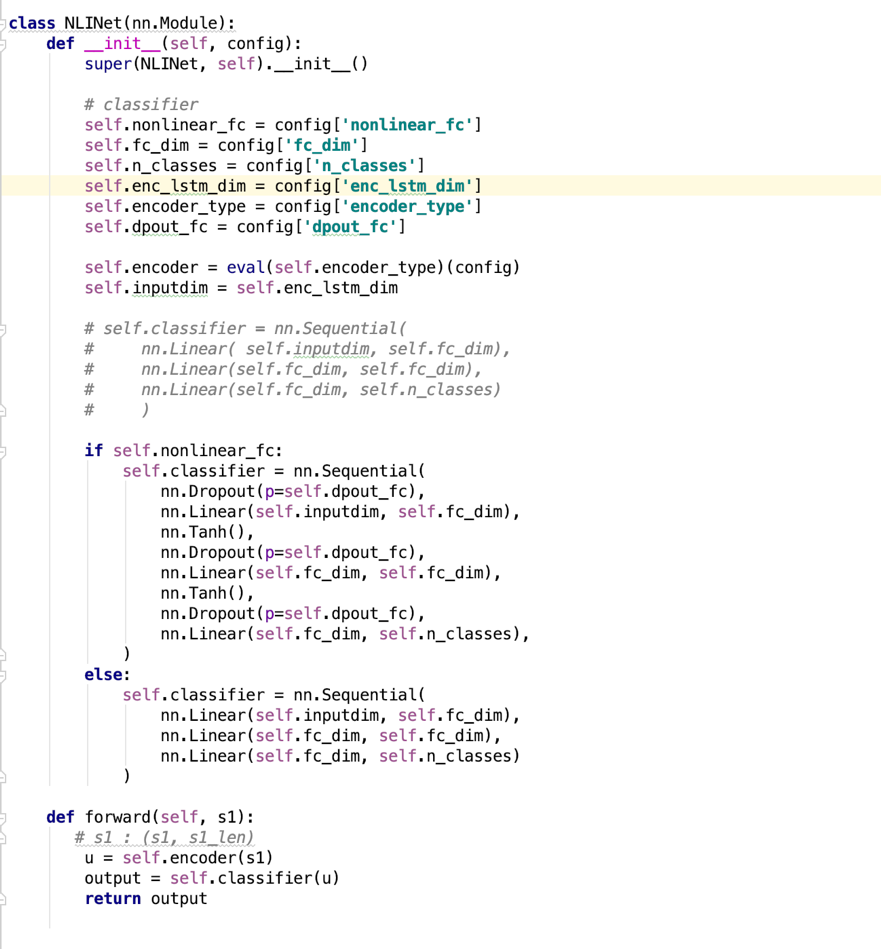
**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 82.2248**

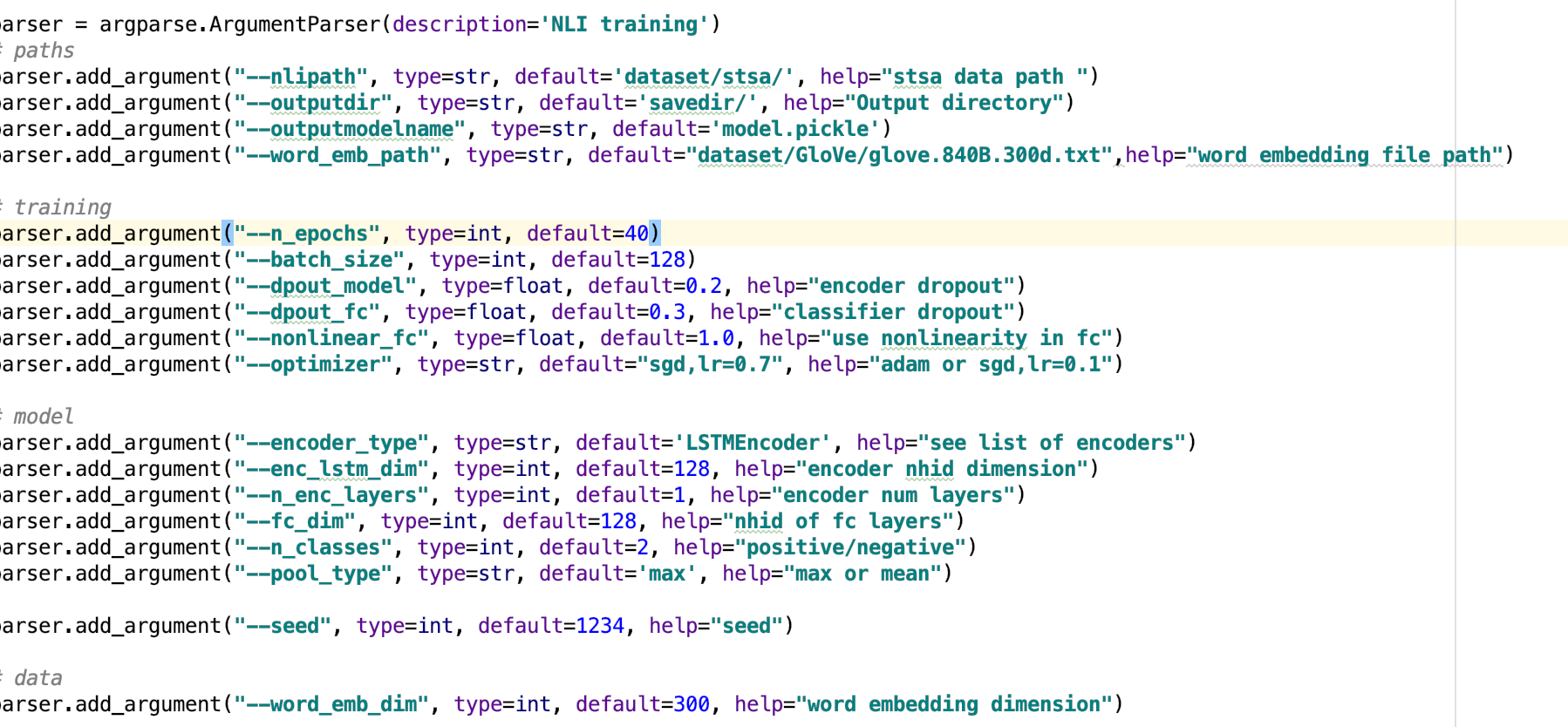
**finalgrep : accuracy test : 82.9215**

**Sol. 3) Have you made structure level modifications of the model : Yes**

**Structural changes in Model.py class for modification and addition of nonlinear and linear layers while classification in neural network :**



**Changes in Train.py class to update the batch-size and adding parameters for supporting and changing dimensions, dropout, pool type, fc-dim, optimizer and batch size parameters of model :**



**1) Changing the model by adding new non-linear layers and 30% dropout :**

**Result :** Training time increases as model is non-linear and dropout and activation layers are added in classifier

**Namespace(batch\_size=64, dpout\_fc=1, dpout\_model=0.0, enc\_lstm\_dim=128, encoder\_type='LSTMEncoder', fc\_dim=64, n\_classes=2, n\_enc\_layers=1, n\_epochs=50, nlipath='dataset/stsa/', nonlinear\_fc=1, optimizer='adam', outputdir='savedir/', outputmodelname='model.pickle', pool\_type='max', seed=1234, word\_emb\_dim=300, word\_emb\_path='dataset/GloVe/glove.840B.300d.txt')**

**\*\* TRAIN DATA : Found 6920 pairs of train sentences.**

**\*\* DEV DATA : Found 872 pairs of dev sentences.**

**\*\* TEST DATA : Found 1821 pairs of test sentences.**

**Found 16517(/17576) words with glove vectors**

**Vocab size : 16517**

**NLINet(**

**(encoder): LSTMEncoder(**

**(enc\_lstm): LSTM(300, 128)**

**)**

**(classifier): Sequential(**

**(0): Dropout(p=1)**

**(1): Linear(in\_features=128, out\_features=64, bias=True)**

**(2): Tanh()**

**(3): Dropout(p=1)**

**(4): Linear(in\_features=64, out\_features=64, bias=True)**

**(5): Tanh()**

**(6): Dropout(p=1)**

**(7): Linear(in\_features=64, out\_features=2, bias=True)**

**)**

**)**

TRAINING : Epoch 1

results : epoch 1 ; loss: 75.46; mean accuracy train : 52.1676

VALIDATION : Epoch 1

togrep : results : epoch 1 ; mean accuracy valid :50.9174

saving model at epoch 1

TEST : Epoch 51

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 83.9174**

**finalgrep : accuracy test : 83.2176**

**2. Changing batch-size and reducing number of epochs with non linear data :**

**Result :** Accuracy increases to 85%

**Namespace(batch\_size=128, dpout\_fc=0.3, dpout\_model=1.0, enc\_lstm\_dim=128, encoder\_type='LSTMEncoder', fc\_dim=64, n\_classes=2, n\_enc\_layers=1, n\_epochs=30, nlipath='dataset/stsa/', nonlinear\_fc=1.0, optimizer='adam', outputdir='savedir/', outputmodelname='model.pickle', pool\_type='max', seed=1234, word\_emb\_dim=300, word\_emb\_path='dataset/GloVe/glove.840B.300d.txt')**

**\*\* TRAIN DATA : Found 6920 pairs of train sentences.**

**\*\* DEV DATA : Found 872 pairs of dev sentences.**

**\*\* TEST DATA : Found 1821 pairs of test sentences.**

**Found 16517(/17576) words with glove vectors**

**Vocab size : 16517**

**/Users/harshverma/anaconda3/lib/python3.6/site-packages/torch/nn/modules/rnn.py:46: UserWarning: dropout option adds dropout after all but last recurrent layer, so non-zero dropout expects num\_layers greater than 1, but got dropout=1.0 and num\_layers=1**

**"num\_layers={}".format(dropout, num\_layers))**

**NLINet(**

**(encoder): LSTMEncoder(**

**(enc\_lstm): LSTM(300, 128, dropout=1.0)**

**)**

**(classifier): Sequential(**

**(0): Dropout(p=0.3)**

**(1): Linear(in\_features=128, out\_features=64, bias=True)**

**(2): Tanh()**

**(3): Dropout(p=0.3)**

**(4): Linear(in\_features=64, out\_features=64, bias=True)**

**(5): Tanh()**

**(6): Dropout(p=0.3)**

**(7): Linear(in\_features=64, out\_features=2, bias=True)**

**)**

**)**

TRAINING : Epoch 1

results : epoch 1 ; loss: 31.69; mean accuracy train : 68.2225

accuracy valid :77.867

results : epoch 21 ; loss: 1.69; mean accuracy train : 99.0607

TEST : Epoch 31

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 84.0596**

**finalgrep : accuracy test : 85.0082**

**3. Increasing dropout and learning rate :**

Namespace(batch\_size=128, dpout\_fc=0.3, dpout\_model=0.2, enc\_lstm\_dim=128, encoder\_type='LSTMEncoder', fc\_dim=128, n\_classes=2, n\_enc\_layers=1, n\_epochs=40, nlipath='dataset/stsa/', nonlinear\_fc=1.0, optimizer='sgd,lr=0.7', outputdir='savedir/', outputmodelname='model.pickle', pool\_type='max', seed=1234, word\_emb\_dim=300, word\_emb\_path='dataset/GloVe/glove.840B.300d.txt')

\*\* TRAIN DATA : Found 6920 pairs of train sentences.

\*\* DEV DATA : Found 872 pairs of dev sentences.

\*\* TEST DATA : Found 1821 pairs of test sentences.

Found 16517(/17576) words with glove vectors

Vocab size : 16517

/Users/harshverma/anaconda3/lib/python3.6/site-packages/torch/nn/modules/rnn.py:46: UserWarning: dropout option adds dropout after all but last recurrent layer, so non-zero dropout expects num\_layers greater than 1, but got dropout=0.2 and num\_layers=1

"num\_layers={}".format(dropout, num\_layers))

**NLINet(**

**(encoder): LSTMEncoder(**

**(enc\_lstm): LSTM(300, 128, dropout=0.2)**

**)**

**(classifier): Sequential(**

**(0): Dropout(p=0.3)**

**(1): Linear(in\_features=128, out\_features=128, bias=True)**

**(2): Tanh()**

**(3): Dropout(p=0.3)**

**(4): Linear(in\_features=128, out\_features=128, bias=True)**

**(5): Tanh()**

**(6): Dropout(p=0.3)**

**(7): Linear(in\_features=128, out\_features=2, bias=True)**

**)**

**)**

TRAINING : Epoch 1

results : epoch 1 ; loss: 38.09; mean accuracy train : 51.8064

TEST : Epoch 41

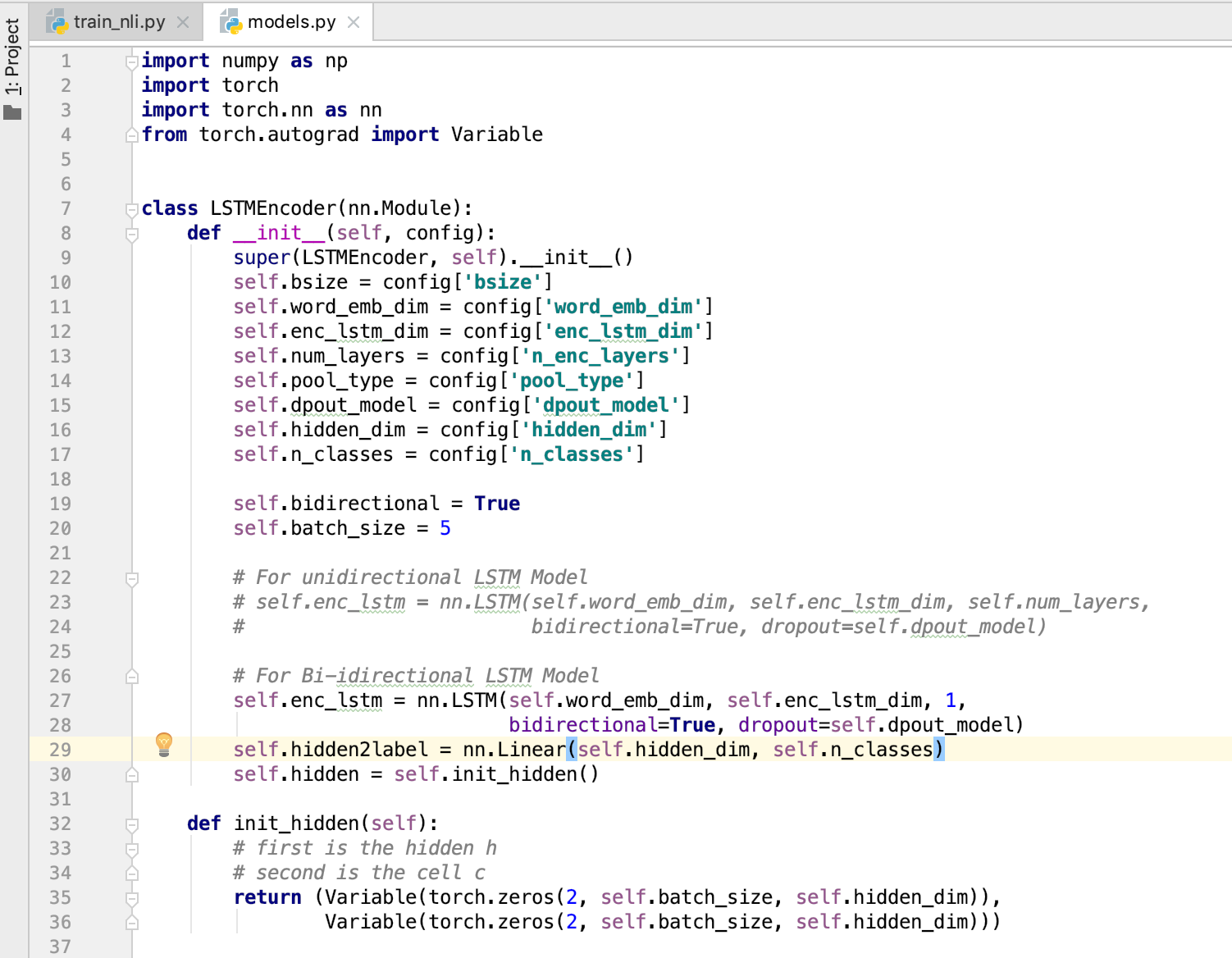
**VALIDATION : Epoch 1000000.0**

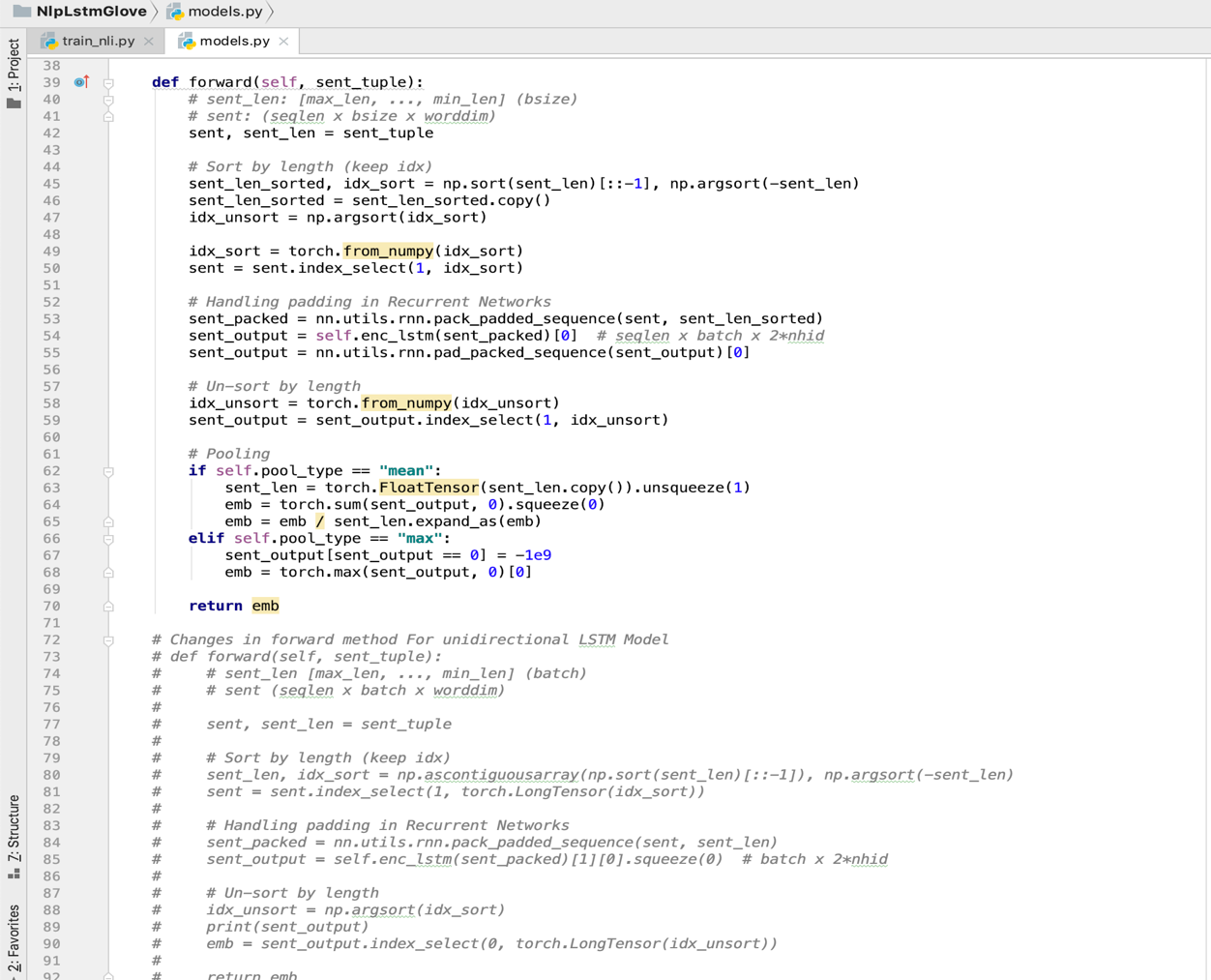
**finalgrep : accuracy valid : 82.4541**

**finalgrep : accuracy test : 83.3059**

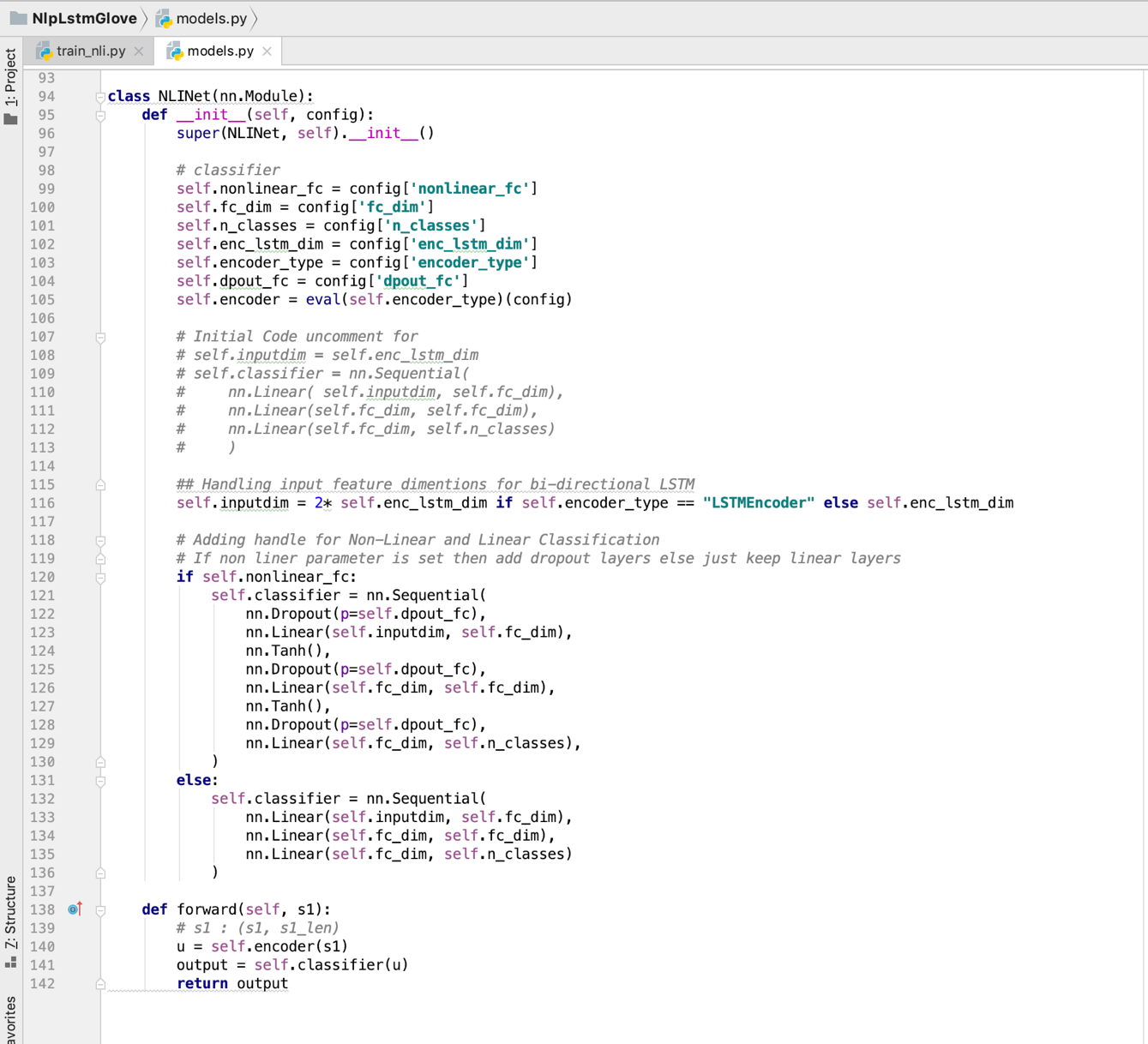
**4) Have you extended the LSTM model to Bi-LSTM model : Yes**

**Attaching the screenshot of changes in code level (Model.py and train.py)**

****

* **Changes in Neural Network to build a support for linear and non Linear Classifier with bi-directional LSTM encoder along with changes in Train.py for hidden layer dimensions.**

****



**Changing to Linear Classification model with Bi-Directional LSTM:**

**Result:**

**Increased accuracy but indeed more time to train the bi-directional model, the model will increase accuracy more if more training data is provided to bi-directional model**

**Namespace(batch\_size=128, dpout\_fc=0.3, dpout\_model=0.2, enc\_lstm\_dim=128, encoder\_type='LSTMEncoder', fc\_dim=128, hidden\_dim=128**

**n\_classes=2, n\_enc\_layers=2, n\_epochs=40, nlipath='dataset/stsa/', nonlinear\_fc=0.0, optimizer='sgd,lr=0.7', outputdir='savedir/', outputmodelname='model.pickle', pool\_type='max', seed=1234, word\_emb\_dim=300, word\_emb\_path='dataset/GloVe/glove.840B.300d.txt')**

**\*\* TRAIN DATA : Found 6920 pairs of train sentences.**

**\*\* DEV DATA : Found 872 pairs of dev sentences.**

**\*\* TEST DATA : Found 1821 pairs of test sentences.**

**Found 16517(/17576) words with glove vectors**

**Vocab size : 16517**

**/Users/harshverma/anaconda3/lib/python3.6/site-packages/torch/nn/modules/rnn.py:46: UserWarning: dropout option adds dropout after all but last recurrent layer, so non-zero dropout expects num\_layers greater than 1, but got dropout=0.2 and num\_layers=1**

**"num\_layers={}".format(dropout, num\_layers))**

**NLINet(**

**(encoder): LSTMEncoder(**

**(enc\_lstm): LSTM(300, 128, dropout=0.2, bidirectional=True)**

**)**

**(classifier): Sequential(**

**(0): Linear(in\_features=256, out\_features=128, bias=True)**

**(1): Linear(in\_features=128, out\_features=128, bias=True)**

**(2): Linear(in\_features=128, out\_features=2, bias=True)**

TRAINING : Epoch 1

results : epoch 1 ; loss: 37.81; mean accuracy train : 54.6676

VALIDATION : Epoch 1

togrep : results : epoch 1 ; mean accuracy valid :73.5092

saving model at epoch 1

TEST : Epoch 41

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 82.9128**

**finalgrep : accuracy test : 84.2943**

**Changing to Non Linear Classification model with Bi-Directional LSTM:**

**Result:**

**Increased accuracy but indeed more time to train the bi-directional model, the model will increase accuracy more if more training data is provided to bi-directional model**

**Namespace(batch\_size=128, dpout\_fc=0.3, dpout\_model=0.2, enc\_lstm\_dim=128, encoder\_type='LSTMEncoder', fc\_dim=128, hidden\_dim=64, n\_classes=2, n\_enc\_layers=1, n\_epochs=30, nlipath='dataset/stsa/', nonlinear\_fc=1.0, optimizer='adam', outputdir='savedir/', outputmodelname='model.pickle', pool\_type='max', seed=1234, word\_emb\_dim=300, word\_emb\_path='dataset/GloVe/glove.840B.300d.txt')**

**\*\* TRAIN DATA : Found 6920 pairs of train sentences.**

**\*\* DEV DATA : Found 872 pairs of dev sentences.**

**\*\* TEST DATA : Found 1821 pairs of test sentences.**

**Found 16517(/17576) words with glove vectors**

**Vocab size : 16517**

**/Users/harshverma/anaconda3/lib/python3.6/site-packages/torch/nn/modules/rnn.py:46: UserWarning: dropout option adds dropout after all but last recurrent layer, so non-zero dropout expects num\_layers greater than 1, but got dropout=0.2 and num\_layers=1**

**"num\_layers={}".format(dropout, num\_layers))**

**NLINet(**

**(encoder): LSTMEncoder(**

**(enc\_lstm): LSTM(300, 128, dropout=0.2, bidirectional=True)**

**)**

**(classifier): Sequential(**

**(0): Dropout(p=0.3)**

**(1): Linear(in\_features=256, out\_features=128, bias=True)**

**(2): Tanh()**

**(3): Dropout(p=0.3)**

**(4): Linear(in\_features=128, out\_features=128, bias=True)**

**(5): Tanh()**

**(6): Dropout(p=0.3)**

**(7): Linear(in\_features=128, out\_features=2, bias=True)**

**)**

**)**

**Final Accuracy after 30 epochs** :

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 83.6009**

**finalgrep : accuracy test : 85.777**

**Further Extension: Adding Hidden Layers to Bi-Directional LSTM Encoders :**

**Non-Linear model in classification with hidden layer in LSTM encoder.**

**Result:**

**Increased accuracy as hidden layers are added in encoder but indeed more time to train the bi-directional model, the model will increase accuracy more if more training data is provided to bi-directional model**

**Namespace(batch\_size=128, dpout\_fc=0.3, dpout\_model=0.2, enc\_lstm\_dim=128, encoder\_type='LSTMEncoder', fc\_dim=128, hidden\_dim=64, n\_classes=2, n\_enc\_layers=1, n\_epochs=30, nlipath='dataset/stsa/', nonlinear\_fc=1.0, optimizer='adam', outputdir='savedir/', outputmodelname='model.pickle', pool\_type='max', seed=1234, word\_emb\_dim=300, word\_emb\_path='dataset/GloVe/glove.840B.300d.txt')**

**\*\* TRAIN DATA : Found 6920 pairs of train sentences.**

**\*\* DEV DATA : Found 872 pairs of dev sentences.**

**\*\* TEST DATA : Found 1821 pairs of test sentences.**

**Found 16517(/17576) words with glove vectors**

**Vocab size : 16517**

**/Users/harshverma/anaconda3/lib/python3.6/site-packages/torch/nn/modules/rnn.py:46: UserWarning: dropout option adds dropout after all but last recurrent layer, so non-zero dropout expects num\_layers greater than 1, but got dropout=0.2 and num\_layers=1**

**"num\_layers={}".format(dropout, num\_layers))**

**NLINet(**

**(encoder): LSTMEncoder(**

**(enc\_lstm): LSTM(300, 128, dropout=0.2, bidirectional=True)**

**(hidden2label): Linear(in\_features=64, out\_features=2, bias=True)**

**)**

**(classifier): Sequential(**

**(0): Dropout(p=0.3)**

**(1): Linear(in\_features=256, out\_features=128, bias=True)**

**(2): Tanh()**

**(3): Dropout(p=0.3)**

**(4): Linear(in\_features=128, out\_features=128, bias=True)**

**(5): Tanh()**

**(6): Dropout(p=0.3)**

**(7): Linear(in\_features=128, out\_features=2, bias=True)**

**)**

**)**

TRAINING : Epoch 1

results : epoch 1 ; loss: 29.58; mean accuracy train : 70.8382

VALIDATION : Epoch 1

togrep : results : epoch 1 ; mean accuracy valid :78.8991

saving model at epoch 1

TEST : Epoch 31

**VALIDATION : Epoch 1000000.0**

**finalgrep : accuracy valid : 84.1743**

**finalgrep : accuracy test : 84.8435**