Analysis: Adv. NLP Assignment 1

Bhaskar Joshi

2019111002

```
Link to the codebase:
https://iiitaphyd-my.sharepoint.com/:f:/g/personal/bhaskar_joshi_research_iiit_ac_in/Es07qyfEMeZElb14TwSrz2kBs-inpruG5lRfiYEx7LNubg?e=qdCbG
```

https://iiitaphyd-my.sharepoint.com/:f:/g/personal/bhaskar_joshi_research_iiit_ac_in/Es07qyfEMeZElb14TwSrz2kBs-inpruG5lRfiYEx7LNubg?e=qdCbGg

Structure

```
- Q1.py
- Q2.py
- Other output files (generated accordingly)
- rnn.py
- gru.py
```

Bonus parts have also been done!!

Running command:

```
For Q1:
python Q1.py >> Q1_log.py

For Q2: Change train to True, for training and val to 0 for RNN
python Q2.py >> Q2_log_rnn.py

For Q2: Change train to True for training and val to 1 for GRU
python Q2.py >> Q2_log_gru.py
```

```
train = False
var=0 # 0 for RNN, 1 for GRU
if var==0:
    save_name="model_rnn"
else:
    save_name="model_gru"
```

Preplexity and normalization

$$PP(W) = \sqrt[N]{\frac{1}{P(w_1, w_2, \ldots, w_N)}}$$

$$P(W) = P(w_1, w_2, \ldots, w_N) = P(w_1)P(w_2)\ldots P(w_N) = \prod_{i=1}^{N} P(w_i)$$

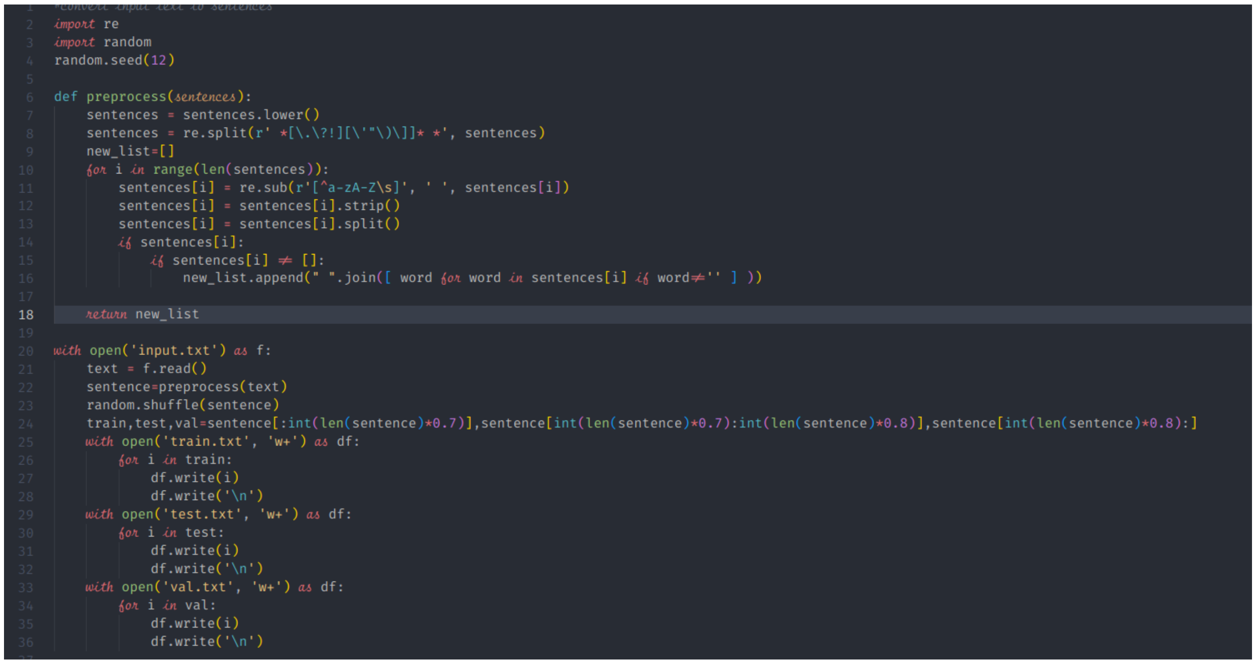
Most Optimal Hyperparameter:

Optimizer: Adam

Learning rate: 1e-3

Performed well on: word2vec

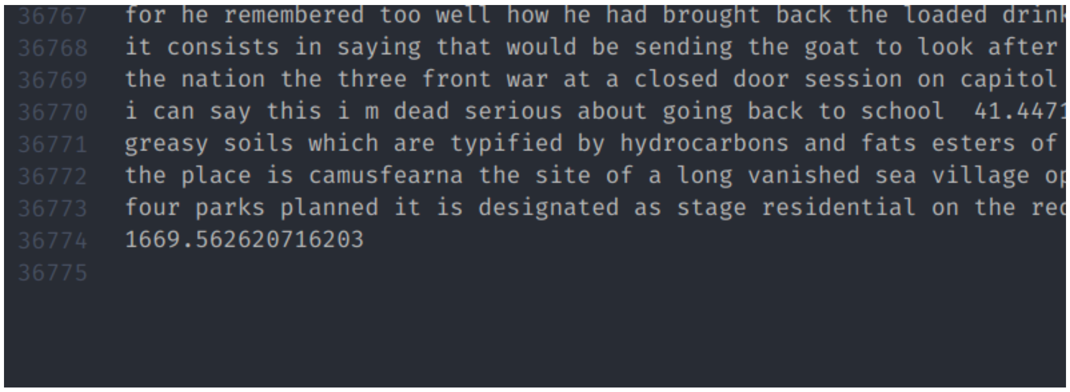
Preprocessing: (in clean_input.py)



```
#convert input text to sentences
import re
import random
random.seed(12)

def preprocess(sentences):
    sentences = sentences.lower()
    sentences = re.split(r' *[\.\?!][\'"\)\]]* *', sentences)
    new_list=[]
    for i in range(len(sentences)):
        sentences[i] = re.sub(r'[^a-zA-Z\s]', ' ', sentences[i])
        sentences[i] = sentences[i].strip()
        sentences[i] = sentences[i].split()
        if sentences[i]:
            if sentences[i] ≠ []:
                new_list.append(" ".join([ word for word in sentences[i] if word≠'' ] ))

    return new_list

with open('input.txt') as f:
    text = f.read()
    sentence=preprocess(text)
    random.shuffle(sentence)
    train,test,val=sentence[:int(len(sentence)*0.7)],sentence[int(len(sentence)*0.7):int(len(sentence)*0.8)],sentence[int(len(sentence)*0.8):]
    with open('train.txt', 'w+') as df:
        for i in train:
            df.write(i)
            df.write('\n')
    with open('test.txt', 'w+') as df:
        for i in test:
            df.write(i)
            df.write('\n')
    with open('val.txt', 'w+') as df:
        for i in val:
            df.write(i)
            df.write('\n')
```

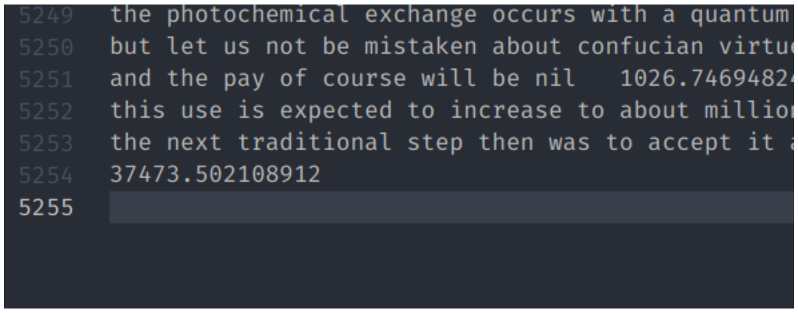
LM1

Train perplexities:



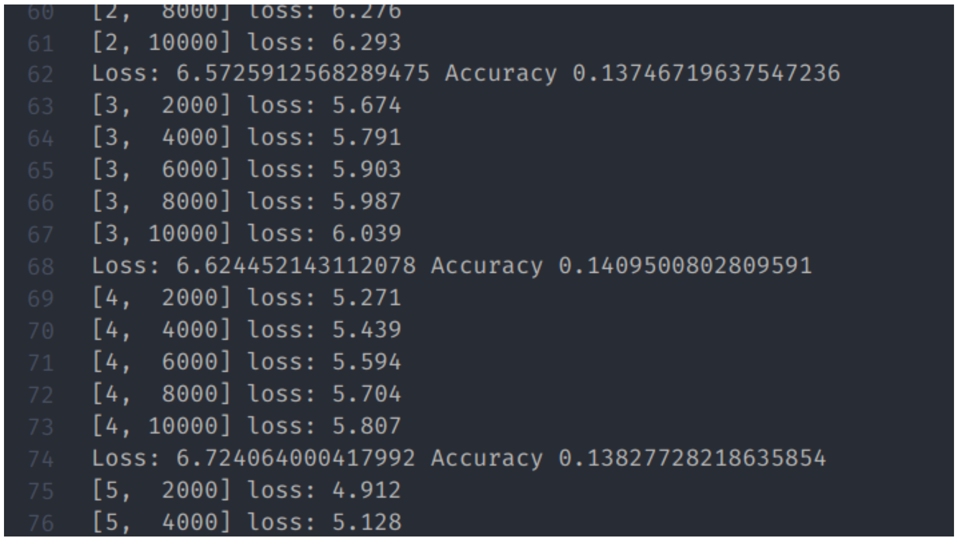
```
for he remembered too well how he had brought back the loaded drin
it consists in saying that would be sending the goat to look after
the nation the three front war at a closed door session on capitol
i can say this i m dead serious about going back to school  41.447
greasy soils which are typified by hydrocarbons and fats esters of
the place is camusfearna the site of a long vanished sea village o
four parks planned it is designated as stage residential on the re
1669.562620716203
```

Test Preplexities:



```
the photochemical exchange occurs with a quantum
but let us not be mistaken about confucian virtu
and the pay of course will be nil   1026.7469482
this use is expected to increase to about millio
the next traditional step then was to accept it
37473.502108912
```

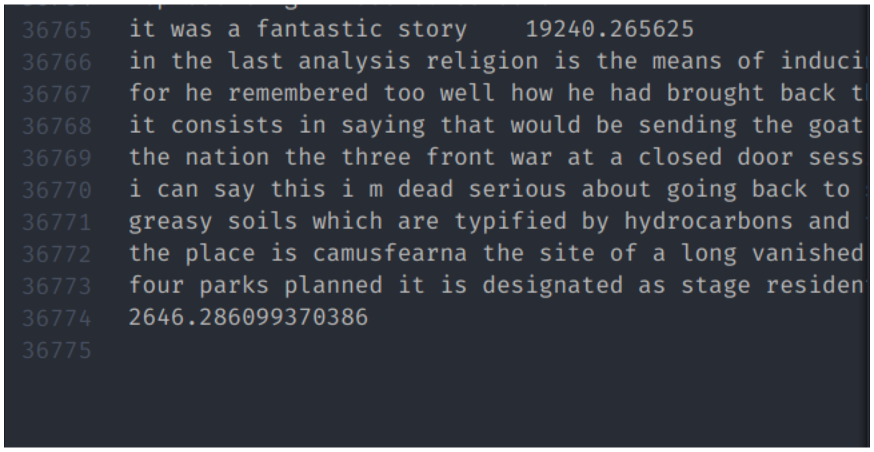
Accuracy:



```
[2,  8000] loss: 6.276
[2, 10000] loss: 6.293
Loss: 6.5725912568289475 Accuracy 0.13746719637547236
[3,  2000] loss: 5.674
[3,  4000] loss: 5.791
[3,  6000] loss: 5.903
[3,  8000] loss: 5.987
[3, 10000] loss: 6.039
Loss: 6.624452143112078 Accuracy 0.1409500802809591
[4,  2000] loss: 5.271
[4,  4000] loss: 5.439
[4,  6000] loss: 5.594
[4,  8000] loss: 5.704
[4, 10000] loss: 5.807
Loss: 6.724064000417992 Accuracy 0.13827728218635854
[5,  2000] loss: 4.912
[5,  4000] loss: 5.128
```

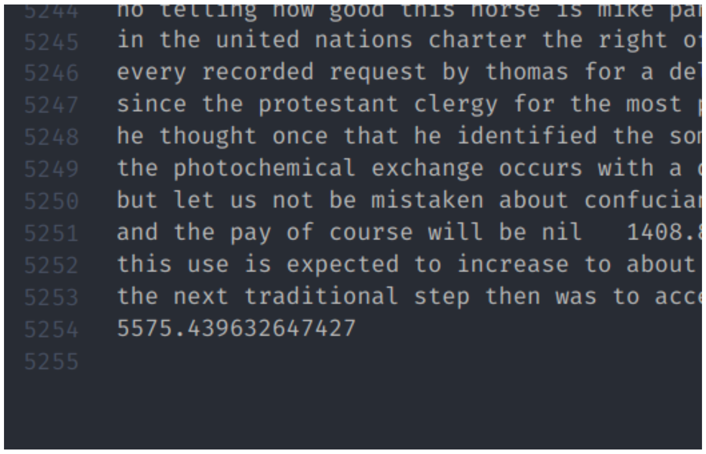
LM2 : RNN

Train:



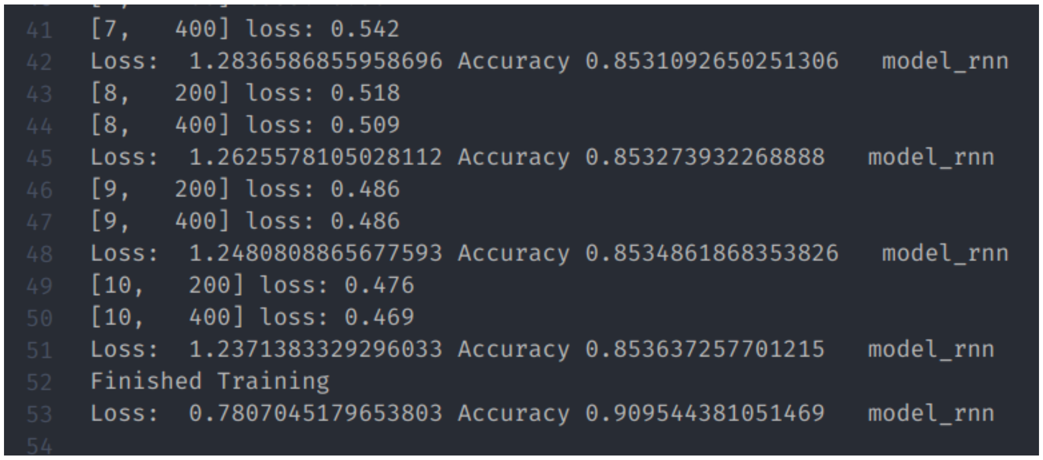
```
it was a fantastic story    19240.265625
in the last analysis religion is the means of induci
for he remembered too well how he had brought back t
it consists in saying that would be sending the goat
the nation the three front war at a closed door sess
i can say this i m dead serious about going back to
greasy soils which are typified by hydrocarbons and
the place is camusfearna the site of a long vanished
four parks planned it is designated as stage residen
2646.286099370386
```

Test:

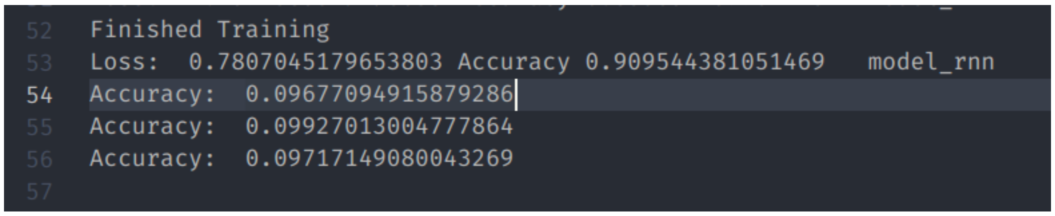


```
no telling how good this horse is mike pa
in the united nations charter the right o
every recorded request by thomas for a de
since the protestant clergy for the most
he thought once that he identified the so
the photochemical exchange occurs with a
but let us not be mistaken about confucia
and the pay of course will be nil   1408.
this use is expected to increase to about
the next traditional step then was to acc
5575.43963264742 7
```

Accuracy:



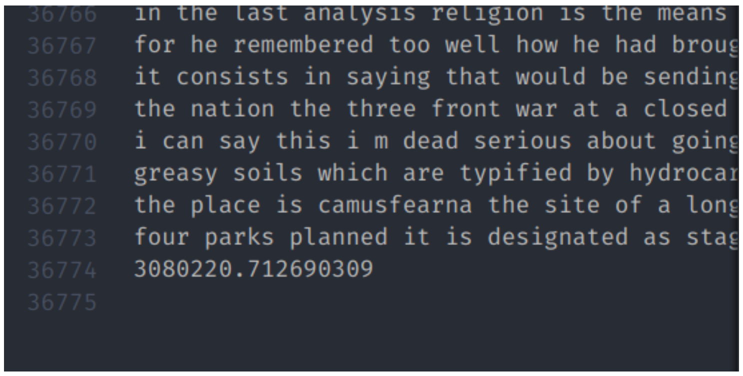
```
[7,   400] loss: 0.542
Loss:  1.2836586855958696 Accuracy 0.8531092650251306   model_rnn
[8,   200] loss: 0.518
[8,   400] loss: 0.509
Loss:  1.2625578105028112 Accuracy 0.853273932268888   model_rnn
[9,   200] loss: 0.486
[9,   400] loss: 0.486
Loss:  1.2480808865677593 Accuracy 0.8534861868353826   model_rnn
[10,   200] loss: 0.476
[10,   400] loss: 0.469
Loss:  1.2371383329296033 Accuracy 0.853637257701215   model_rnn
Finished Training
Loss:  0.7807045179653803 Accuracy 0.909544381051469   model_rnn
```



```
Finished Training
Loss:  0.7807045179653803 Accuracy 0.909544381051469   model_rnn
Accuracy:  0.09677094915879286
Accuracy:  0.09927013004777864
Accuracy:  0.09717149080043269
```

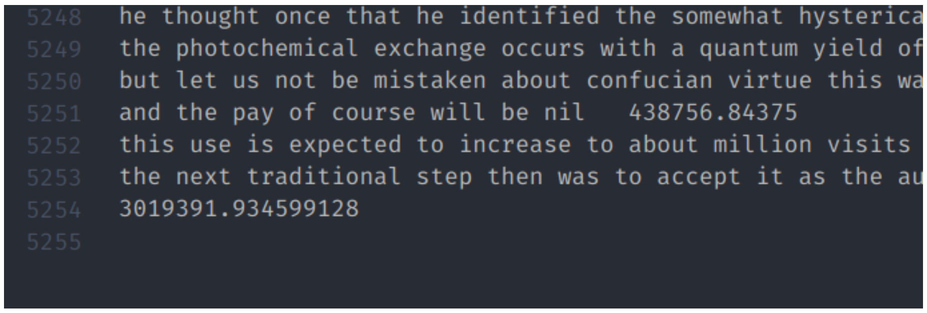
LM2: GRU

Train:



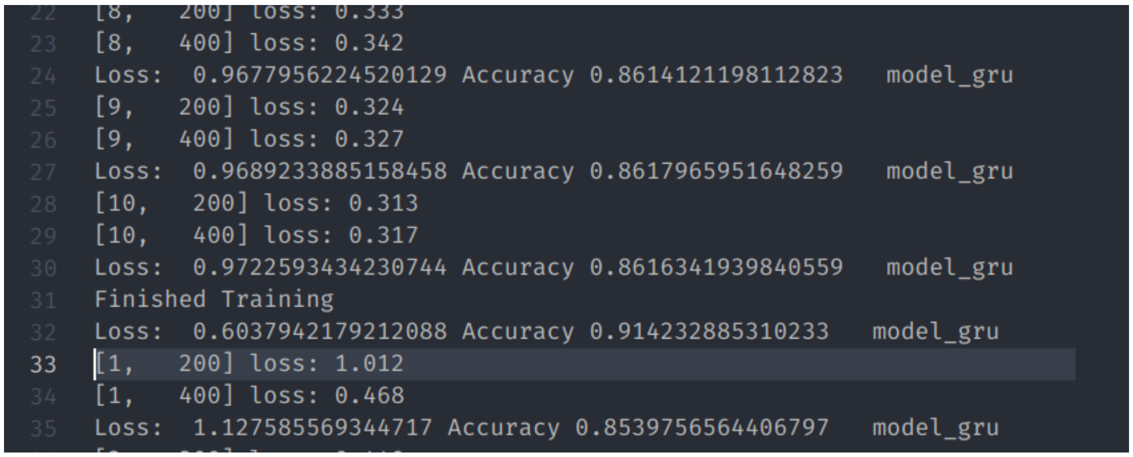
```
in the last analysis religion is the means
for he remembered too well how he had broug
it consists in saying that would be sending
the nation the three front war at a closed
i can say this i m dead serious about going
greasy soils which are typified by hydrocar
the place is camusfearna the site of a long
four parks planned it is designated as stag
3080220.712690309
```

Test:

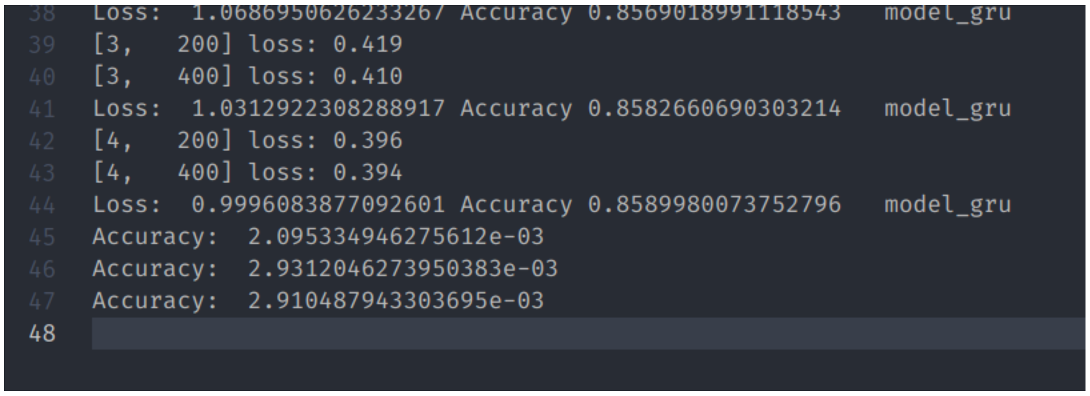


```
he thought once that he identified the somewhat hysterica
the photochemical exchange occurs with a quantum yield of
but let us not be mistaken about confucian virtue this wa
and the pay of course will be nil   438756.84375
this use is expected to increase to about million visits
the next traditional step then was to accept it as the au
3019391.934599128
```

Accuracy:



```
[8,   200] loss: 0.333
[8,   400] loss: 0.342
Loss:  0.9677956224520129 Accuracy 0.8614121198112823   model_gru
[9,   200] loss: 0.324
[9,   400] loss: 0.327
Loss:  0.9689233885158458 Accuracy 0.8617965951648259   model_gru
[10,   200] loss: 0.313
[10,   400] loss: 0.317
Loss:  0.9722593434230744 Accuracy 0.8616341939840559   model_gru
Finished Training
Loss:  0.6037942179212088 Accuracy 0.914232885310233   model_gru
[1,   200] loss: 1.012
[1,   400] loss: 0.468
Loss:  1.127585569344717 Accuracy 0.8539756564406797   model_gru
```



```
Loss:  1.0686950626233267 Accuracy 0.8569018991118543   model_gru
[3,   200] loss: 0.419
[3,   400] loss: 0.410
Loss:  1.0312922308288917 Accuracy 0.8582660690303214   model_gru
[4,   200] loss: 0.396
[4,   400] loss: 0.394
Loss:  0.9996083877092601 Accuracy 0.8589980073752796   model_gru
Accuracy:  2.095334946275612e-03
Accuracy:  2.9312046273950383e-03
Accuracy:  2.910487943303695e-03
```

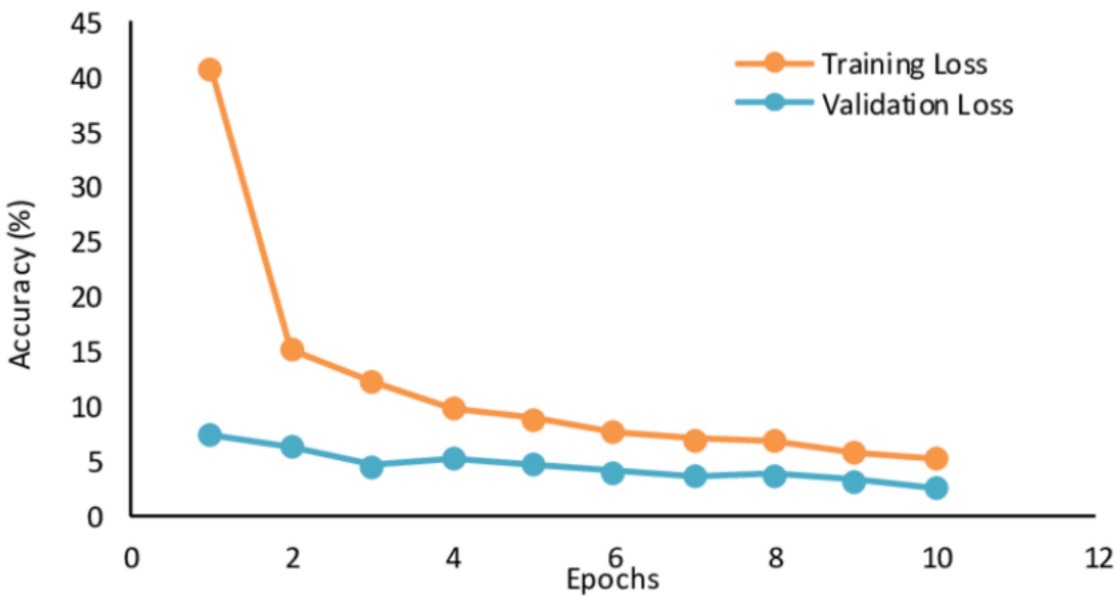
Compare models and tell which is better

Ans: Perplexity Lower and Higher Accuracy

As from the above results LM1 (NNRN) performs well.

RNN LM has accuracy score of ~90.5% but with padding and GRULM had ~91.4% score with padding

Train and validation graph curves for NNLM:



It learns to predict commonly occuring words fast but it hits its limit and the accuracy stalls at around 14%. It is limited by its architecture.

```
Loss: 7.1127519607543945 Accuracy 0.13424917193469277
[1,  2000] loss: 7.156
[1,  4000] loss: 6.790
[1,  6000] loss: 6.692
[1,  8000] loss: 6.639
[1, 10000] loss: 6.601
Loss: 6.570091930403612 Accuracy 0.13380483238014854
[2,  2000] loss: 6.112
[2,  4000] loss: 6.184
```

```
[2,  6000] loss: 6.246
[2,  8000] loss: 6.276
[2, 10000] loss: 6.293
Loss: 6.5725912568289475 Accuracy 0.13746719637547236
[3,  2000] loss: 5.674
[3,  4000] loss: 5.791
[3,  6000] loss: 5.903
[3,  8000] loss: 5.987
[3, 10000] loss: 6.039
Loss: 6.624452143112078 Accuracy 0.1409500802809591
```

For RNN

It learns to predicts padding quite fast and then move on to learn more about the actual sentence. It was able to display some level of long term dependency beyond 3-4 words. Running it on a bigger corpus with more layers and parameters might allow us to model the better.