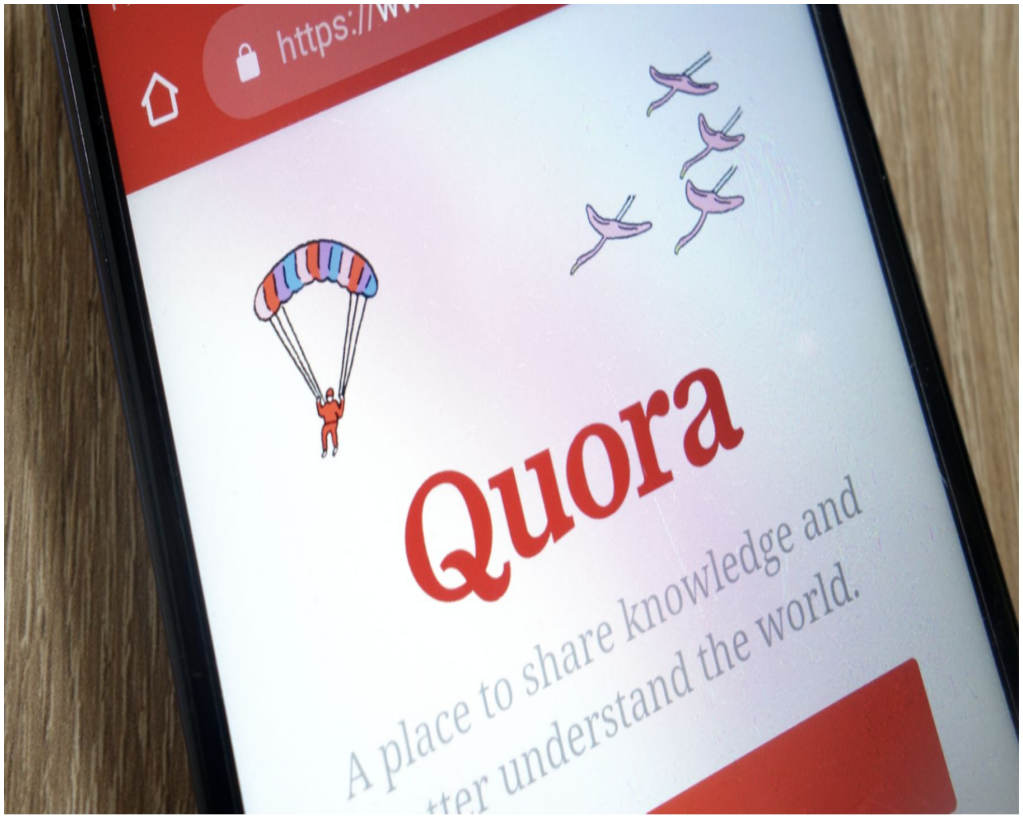
Mini Project V

(Quora Question Similarity Problem)

By - Pooja

Problem

Identify which questions asked on Quora are duplicates of questions that have already been asked.



Objectives and Constraints

The cost of a mis-classification can be very high.

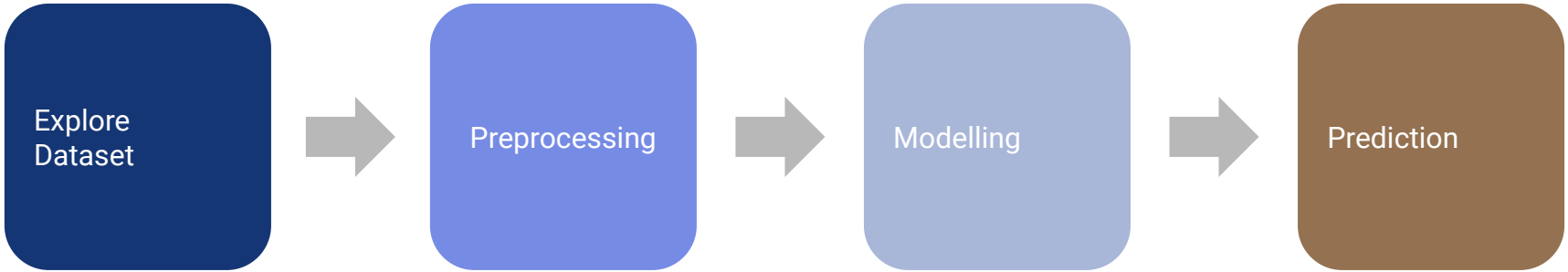
We want a probability of a pair of questions to be duplicates so that you can choose any threshold of choice.

No strict latency concerns.

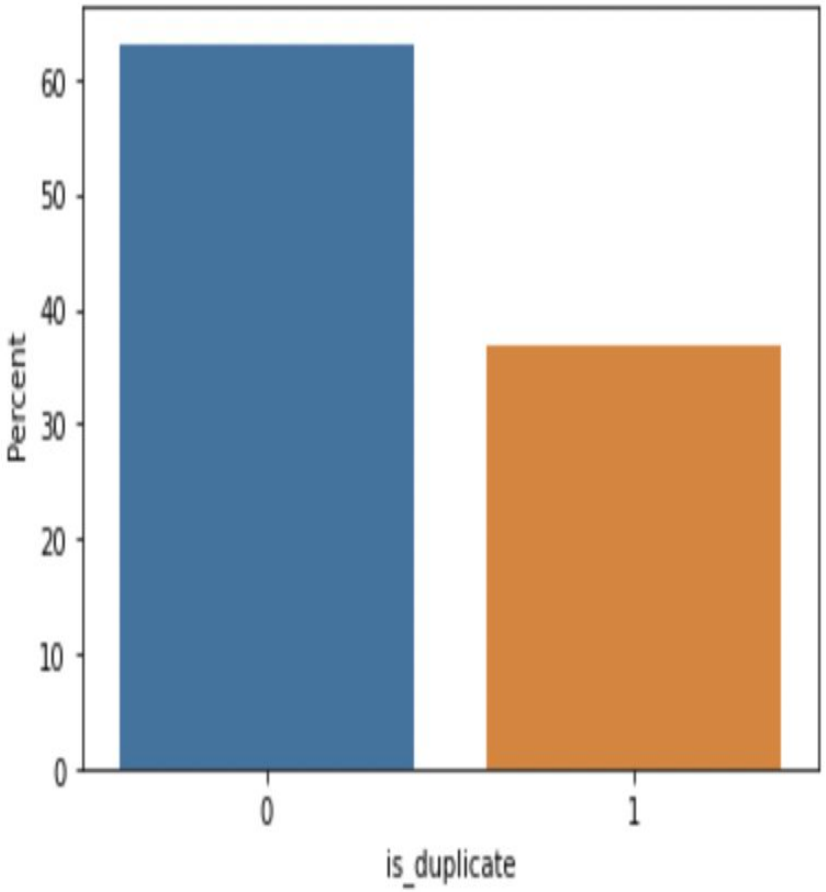
Interpretability is partially important.

Recurrent Neural Network (RNN)

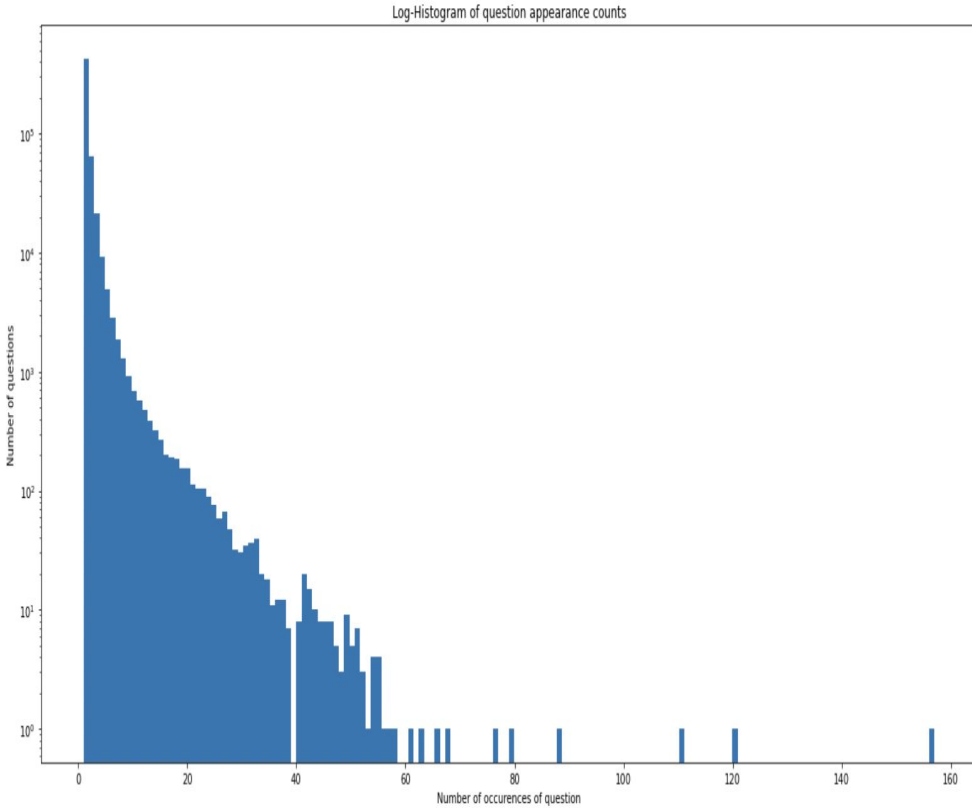
– – Long Short - Term Memory(LSTM)



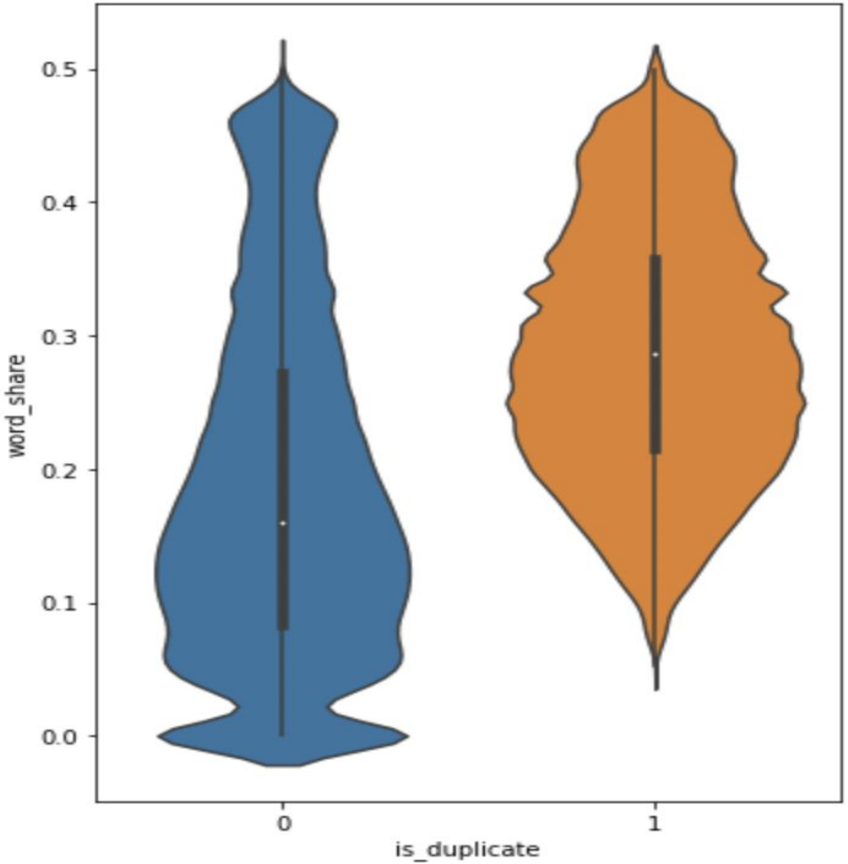
EDA

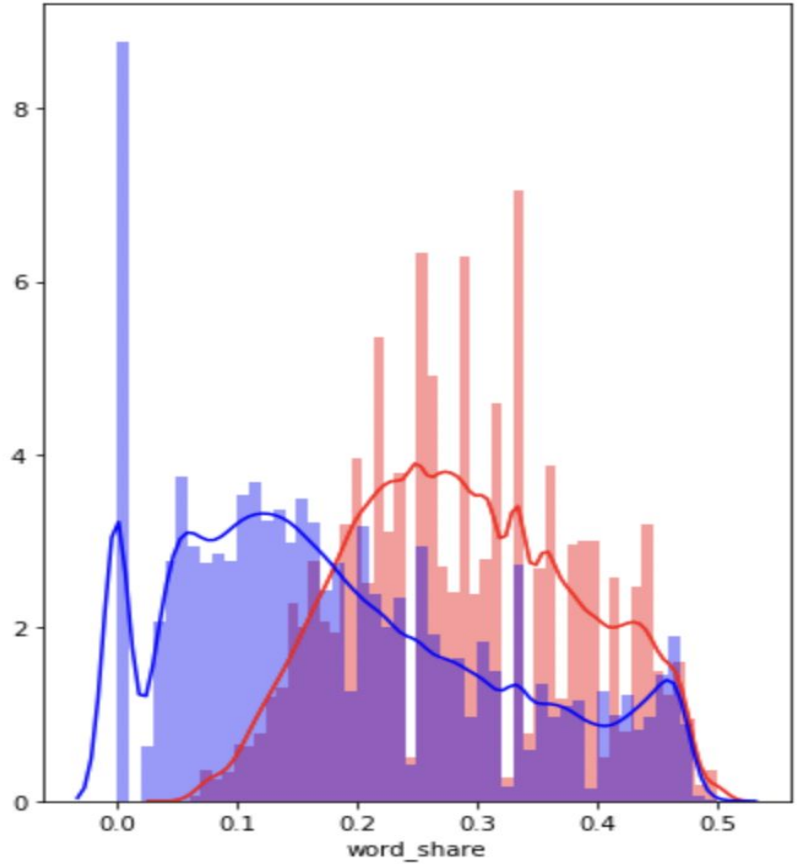


Maximum number of times a single question is repeated: 157

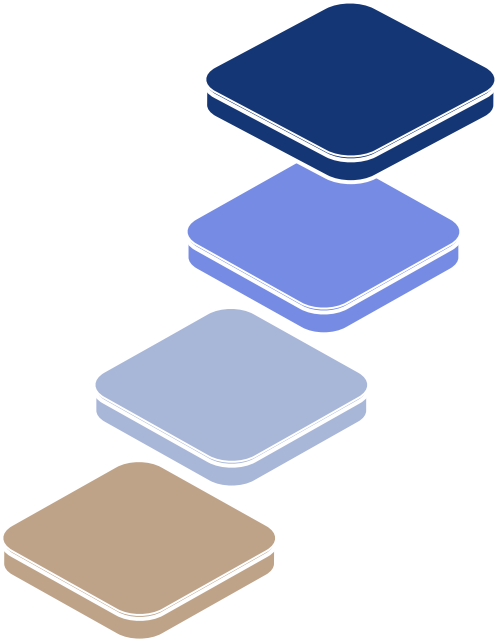


EDA


word_share
0.0
0.1
0.2
0.3
0.4
0.5
0
1
is_duplicate


0
2
4
6
8
0.0
0.1
0.2
0.3
0.4
0.5
word_share

Preprocess Pipeline



Tokenize punctuation and lowercase all tokens.

Clean

Convert each word into a number(word IDs)

Tokenize

Filter sentences.

Filter Length

sequence padding to have the same length of sentence

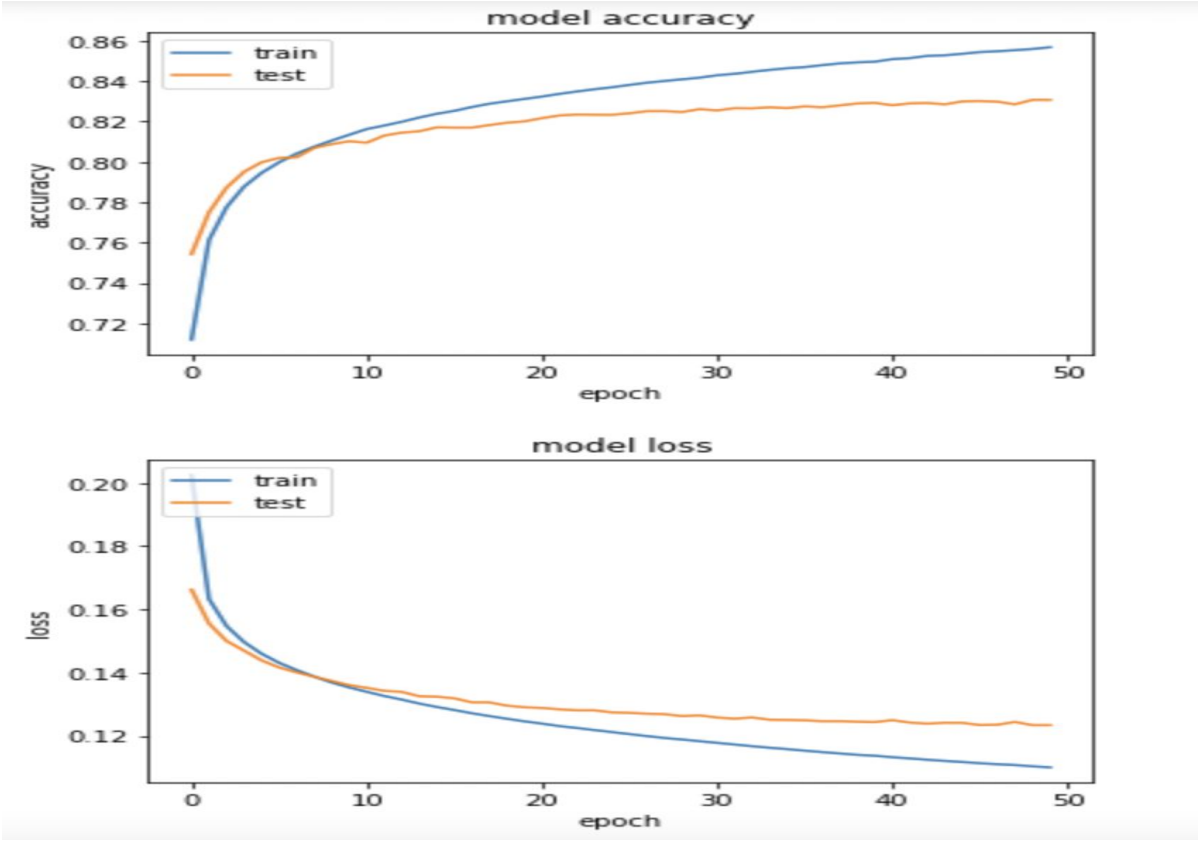
Padding

Model :Sequential

```
input_1 (InputLayer)            [(None, 60)]         0
_____
input_2 (InputLayer)            [(None, 60)]         0
_____
embedding (Embedding)           (None, 60, 300)      60000000    input_1[0][0]
                                                                 input_2[0][0]
_____
lstm (LSTM)                     (None, 50)           70200       embedding[0][0]
                                                                 embedding[1][0]
_____
lambda (Lambda)                 (None, 1)            0           lstm[0][0]
                                                                 lstm[1][0]
==========================================================================================
Total params: 60,070,200
Trainable params: 70,200
Non-trainable params: 60,000,000
```

```
- val_loss: 0.1241 - val_accuracy: 0.8285
Epoch 45/50
178/178 [==============================] - 18s 101ms/step - loss: 0.1112 - accuracy: 0.8544
- val_loss: 0.1241 - val_accuracy: 0.8300
Epoch 46/50
178/178 [==============================] - 18s 100ms/step - loss: 0.1110 - accuracy: 0.8550
- val_loss: 0.1234 - val_accuracy: 0.8301
Epoch 47/50
178/178 [==============================] - 18s 102ms/step - loss: 0.1101 - accuracy: 0.8566
- val_loss: 0.1235 - val_accuracy: 0.8298
Epoch 48/50
178/178 [==============================] - 18s 101ms/step - loss: 0.1105 - accuracy: 0.8558
- val_loss: 0.1244 - val_accuracy: 0.8285
Epoch 49/50
178/178 [==============================] - 18s 100ms/step - loss: 0.1099 - accuracy: 0.8572
- val_loss: 0.1234 - val_accuracy: 0.8307
Epoch 50/50
178/178 [==============================] - 18s 102ms/step - loss: 0.1100 - accuracy: 0.8570
- val_loss: 0.1234 - val_accuracy: 0.8307
```

Model Prediction



Reflection

Deep Learning is very slow.

To create corpus of vectors was also very time consuming.

Can add more features to improve model's performance

Add more layers ar any other model to improve the performance

Thanks!