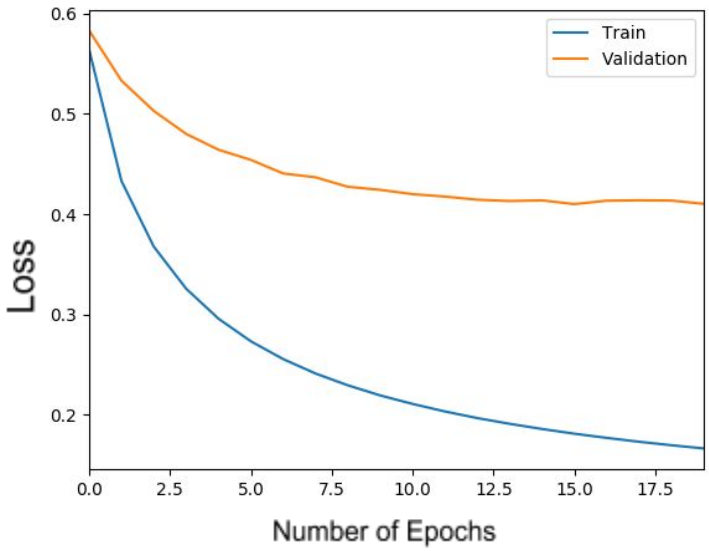
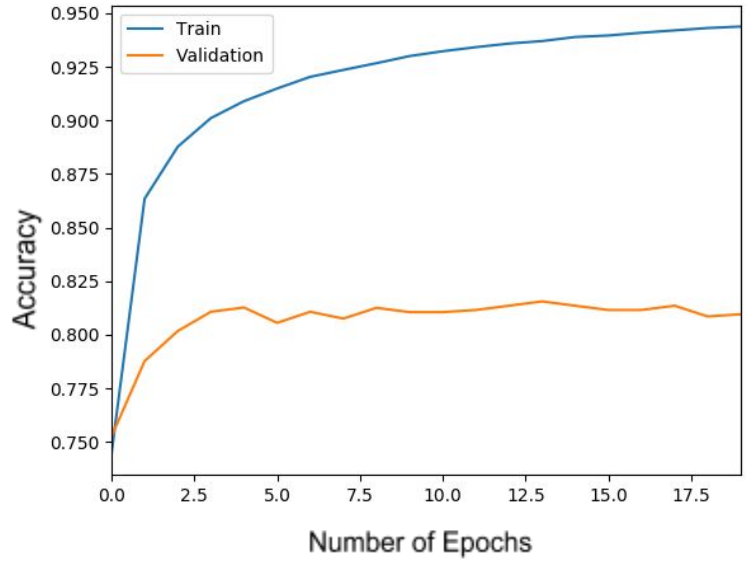
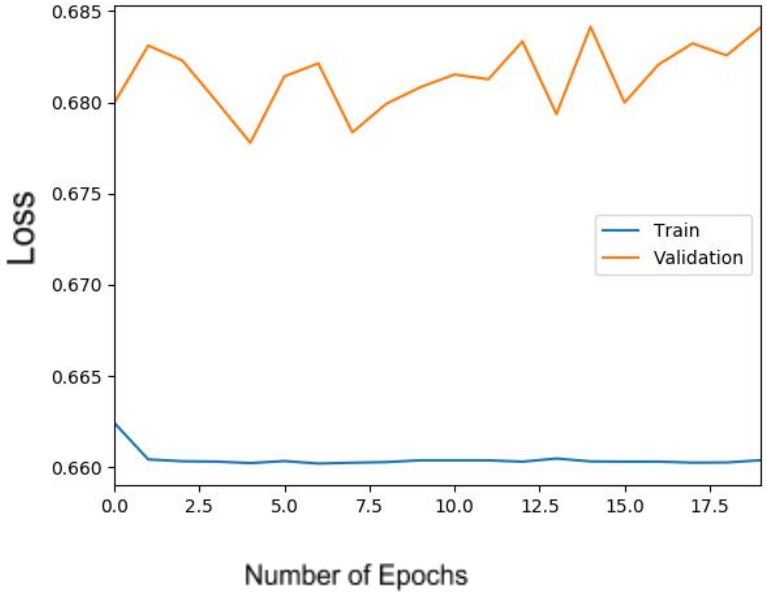
Log-linear results

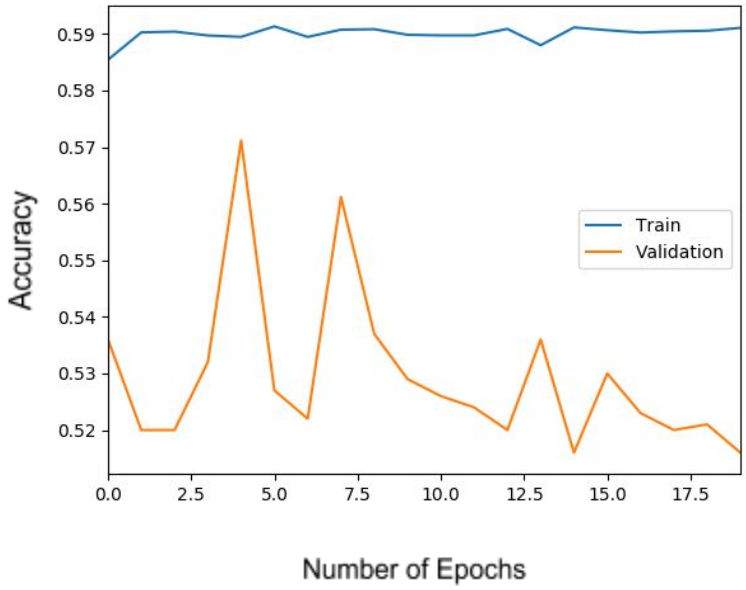
Weight = 0



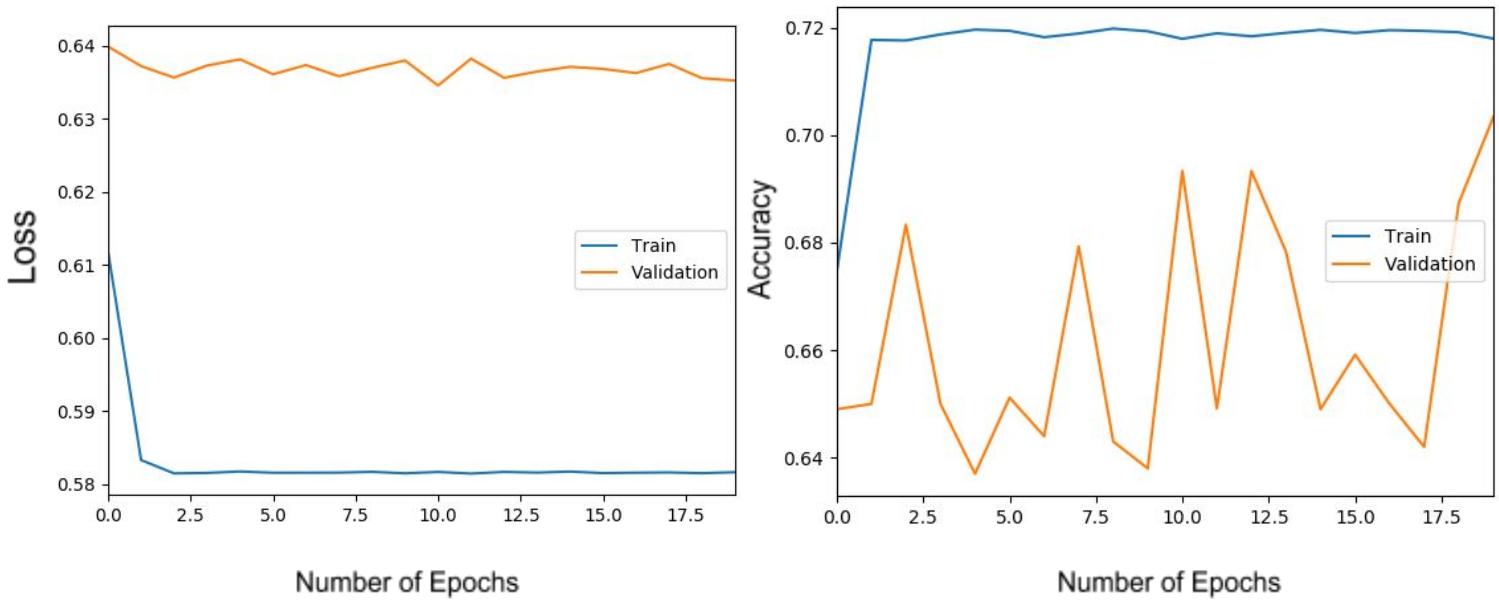


Weight = 0.001





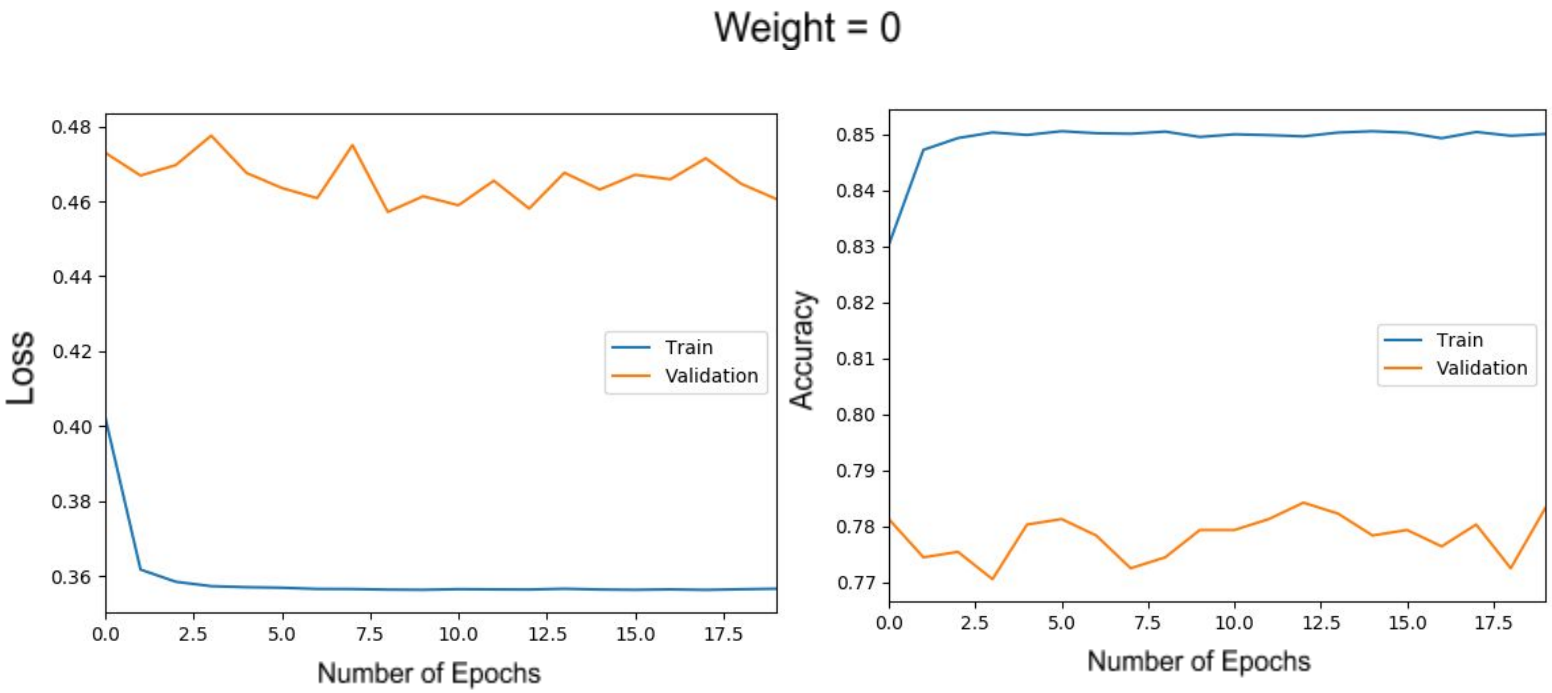
Weight = 0.0001

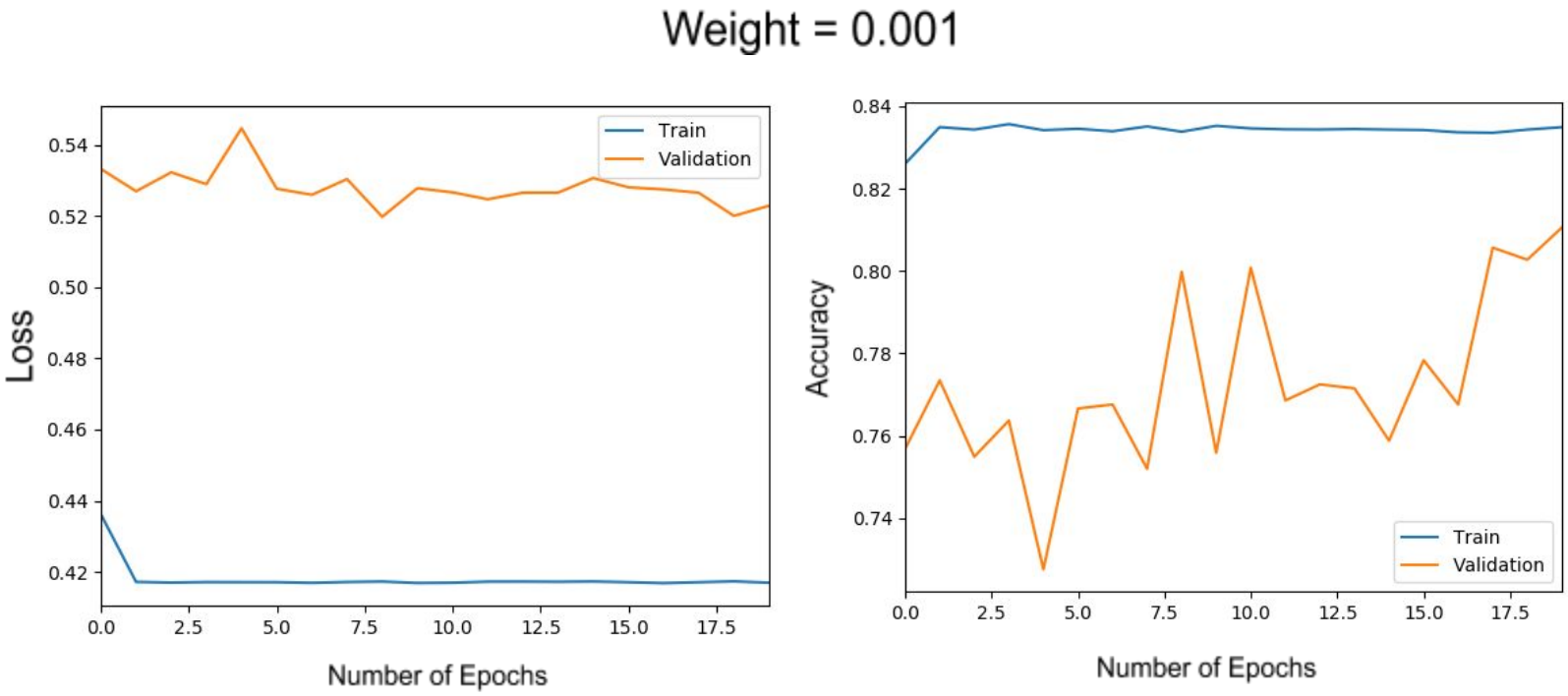


The best weight was as expected 0 value.

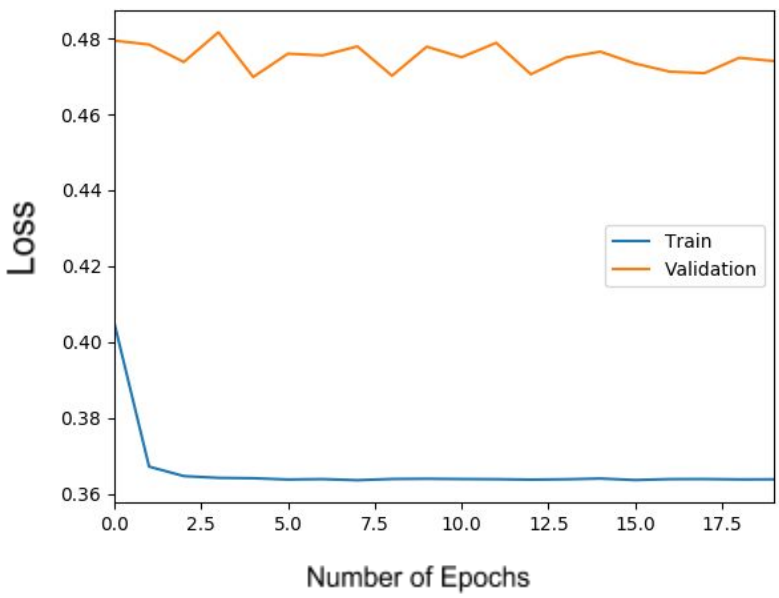
- The test results are:
  - Loss: 0.4139821175453903
  - Accuracy: 0.8118333333333334
- The special subsets accuracies:
  - Negative: 0.5916666666666667
  - Rare: 0.52

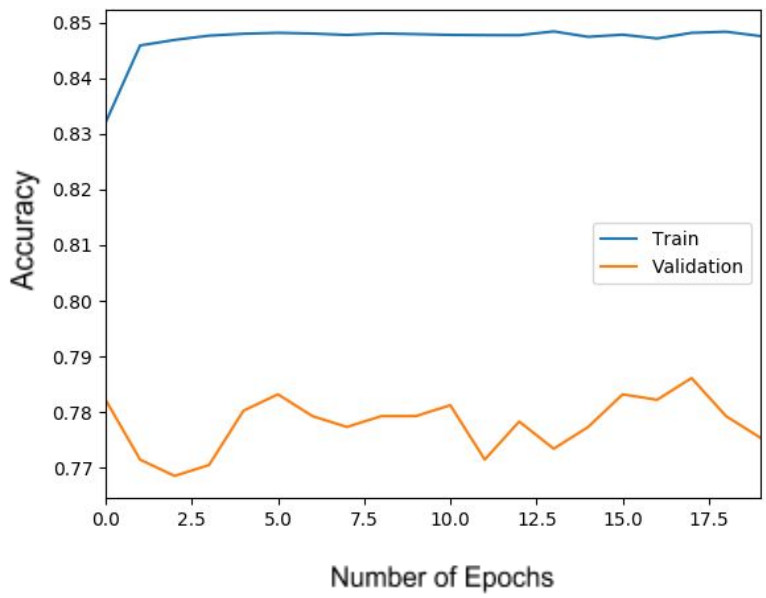
W2V Results


Weight = 0
Loss
Accuracy
Number of Epochs
Number of Epochs
Train
Validation
Train
Validation


Weight = 0.001
Loss
Accuracy
Number of Epochs
Number of Epochs
Train
Validation
Train
Validation

Weight = 0.0001

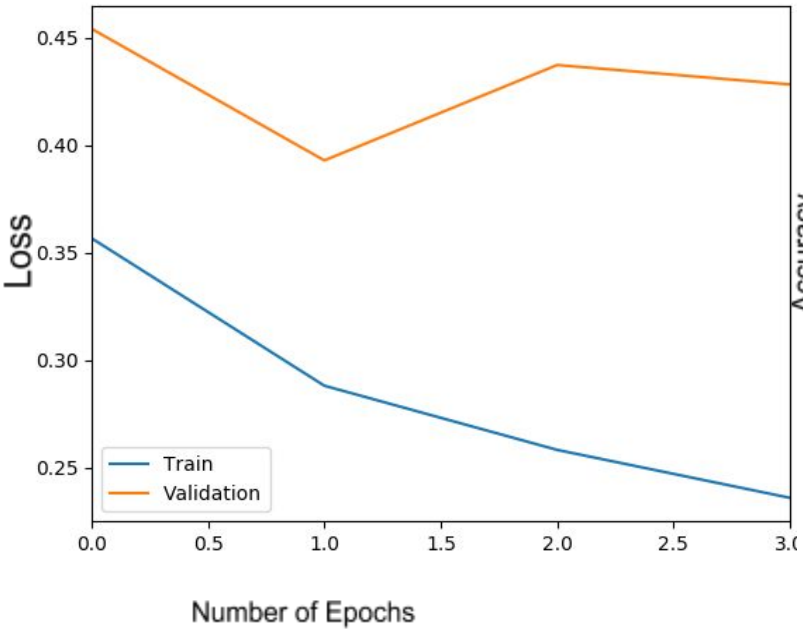


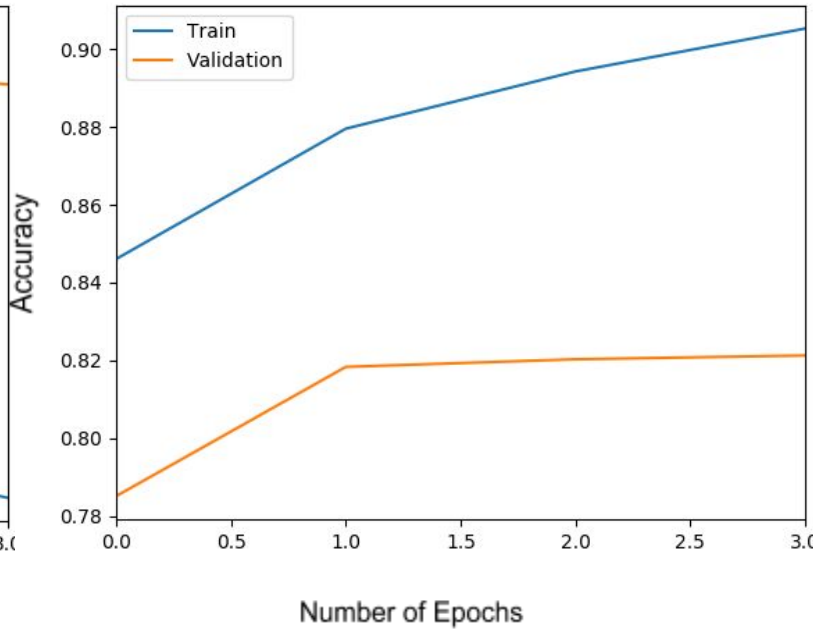


The best weight was as expected 0 value.

- The test results are:
  - Loss: 0.3930098139682201
  - Accuracy: 0.8212890625
- The special subsets accuracies:
  - Negative: 0.6290322580645161
  - Rare: 0.8

LSTM model
Learning rate- 0.001
W = 0.0001
Dropout: 0.5





- The test results are:
  - Loss: 0.30536458030046987
  - Accuracy: 0.869140625
- The special subsets accuracies:
  - Negative: 0.6935483870967742
  - Rare: 0.82

Comparison Questions:

1. As we can see from the results while using weight decays bigger than 0 the W2V method performed better because of the semantic area that the W2V can give us. Without the weight decays we get a slightly better accuracy using the W2V, but both perform well. The model works better because using the W2V model gives us linear dependence between different word vectors, which gives us the semantic area of each word, and as a result we can get better accuracy, especially for special subsets as rare words - getting the semantic area is making them (the rare words) less rare- and we are getting better results.
2. The LSTM model performed better, especially given the small number of epochs we had. The reason for that is that firstly we also used the W2V embedding, so we achieved similar goals that were noted in the first question. The reason the LSTM model itself worked better is that as we said in class the linear log is prone to vanishing gradients, while the LSTM model is far less prone to. We also used dropouts for the LSTM train and that reduced our overfit and brought better results overall.

   Also, the LSTM model is far more complex, being a bi-directional model that considers both ending and the beginning of the sentences and uses a linear layer at every step, and by that each epoch learns more than a simple log linear model can.
3. Between the first two models, the W2V has a great advantage on both subsets, because the linear dependance that we have already mentioned caused the rare and negative words subsets to perform better and get greatly superior accuracy results testing these special subsets.

   When comparing between the LSTM and the log linear models we can clearly see that those subsets tests performed better than both. As mentioned the LSTM uses the W2V, so we expect the results to be at least as good as the ones in W2V. Also, the LSTM model is far more complex, being a bi-directional model that considers both ending and the beginning of the sentences and using a linear layer at every step, and by that each epoch gets more information about the semantic, structure and purpose of the sentence and each word in it. Thus, the special subsets get better results in this model.