NLP – EX3

6 – a

One Hot Log Linear train & validation loss per epoch:

Chart, line chart

Description automatically generated

6 – b

One Hot Log Linear train & validation accuracy per epoch:

Chart, line chart

Description automatically generated

6 – c

One Hot Log Linear Test loss & accuracy:

Loss: **0.6803652548551518** Accuracy: **0.524948024948025**

Accuracy over negated polarity: **0.4838709677419355**

Accuracy over rare words: **0.28**

7 – a

W2V – Log Linear train & validation loss per epoch:

Chart

Description automatically generated

7 – b

W2V – Log Linear train & validation accuracy per epoch:

Chart, line chart

Description automatically generated

7 – c

W2V - LogLinear Test loss & accuracy:

Loss: **0.4735080028892954** Accuracy: **0.8035343035343036**

Accuracy over negated polarity: **0.5967741935483871**

Accuracy over rare words: **0.74**

8 – a

W2V - LSTM train & validation loss per epoch:

Chart, line chart

Description automatically generated

8 – b

W2V - LSTM train & validation accuracy per epoch:

Chart, line chart

Description automatically generated

8 – c

W2V - LSTM Test loss & accuracy:

Loss: **0.3583637079468947** Accuracy: **0.8669438669438669**

Accuracy over negated polarity: **0.6935483870967742**

Accuracy over rare words: **0.82**

9 - 1

*Compare the results (test accuracy, validation accuracy) you've received for the*

*simple log-linear model, and the Word2Vec log-linear model. Which one performs*

*better? Provide a possible explanation for the results you have.*

Word2Vec log linear model performs better than the one hot encoded log linear. A possible explanation for this difference could be the fact that w2v embedding can express word similarities using common neighboring words, thus giving sentiment also to less frequent / previously unseen words, with higher accuracy in comparison to the one hot log linear model.

9 – 2

*Compare the latter results with the results of the LSTM model. Which one performs*

*better? Provide an explanation for the results you received.*

The LSTM model performs better than the previous two model for two main reasons:

1. It uses W2V word embedding, thus gaining the benefits of word similarities, as did the W2V log linear model.
2. It has temporal representation for sentences, which enables to take into consideration previous word sentiments (and forthcoming word sentiments in our Bidirectional model) when determining the sentiment of a given word in a sentence, and as a result, predicting a more accurate sentence sentiment.

9 – 3

*Compare the results that all the models had on the 2 special subsets of sentences we've provided you. For each subset, state the model that has the highest result (and the lowest result) and provide a possible explanation for these results.*

The results:

**One Hot Log Linear**:

Accuracy for negated: 0.4838709677419355

Accuracy for rare words: 0.28

**W2V Log Linear**:

Accuracy for negated: 0.5967741935483871

Accuracy for rare words: 0.74

**W2V LSTM**:

Accuracy for negated: 0.6935483870967742

Accuracy for rare words: 0.82

**Best model for negated words** **is LSTM model**, most likely because of temporal memory during

Sentiment evaluation: if we evaluate the sentence ‘Not good’, ‘good’s sentiment will be impacted by ‘Not’s sentiment.

On the other hand, in the **worst preforming model for rare words – one hot log linear** -

the sentence sentiment will simply be determined by the portion of words with largest sentiment. So, if ‘Not’s negative sentiment is relatively smaller than ‘Good’s positive sentiment, the sentence will be classified as positive.

**Best model for rare words** is once again LSTM. This is likely because of the W2V embedding that relies on neighboring words for sentiment analysis, in addition to the short- and long-term previous word sentiments that are considered during the evaluation of a rare word’s sentiment.

**The worst model for rare words is the log linear model**, likely because of scarce learned examples over rare words, and the lack of ability to accommodate word similarity.

Interestingly, 0.28 is quite low, and if we negate the model’s decision over rare words its performance could improve. We could not decide for certain why this is the case, but perhaps it reflects some ununiform distribution of the sentiments in sampled rare word sentences.