Hw2 Report

Name: HSU, Chia hong; Stid: 20562937

Implement the training function, and report the loss value (at the end of the training) of 5 different hyper-parameter configuration. Use a separate file named main.py.

Implement the two evaluation:
• Word Similarity: implement cosine similarity and 5 couple of words score.
• Sentiment Classifier: use the learned embedding matrix for assignment 1 and submit the prediction results of twitter-sentiment-testset.csv file again.

Word bank selection: {nice, excellent, bad, poor, beautiful, ugly, love, like, long, longer, short, shorter}. We expect the pair-wise similarity magnitude among all positive words or all negative words have positive values, such as "nice", "excellent", "beautiful", etc.. We also expect the embedding to learn the significance of comparative adjectives with respect to normal adjectives, such as "long", "longer".

Experiment A

Epoch	Learning Rate	Embedding Dim	Batch Size	Window Size
20	0.1	300	500	2

Loss = 244398.83091722841

Word similarity

Nice,excellent	0.0976	Correct
Nice,bad	0.0191	Correct
Bad,poor	0.0713	Correct
Nice,poor	-0.0054	Correct
Beautiful,nice	-0.0257	Sentiment not captured
Ugly,nice	-0.0236	Sentiment not captured
Love,like	-0.0270	Sentiment not captured
Long,longer	-0.0034	Sentiment not captured
Short,shorter	0.0270	Sentiment not captured
Long,short	-0.0382	Correct
Longer,shorter	0.0974	Sentiment not captured

Sentiment Classifier:

Best dev accuracy: 65.475%, training accuracy: 67.8%

Experiment B

Epoch	Learning Rate	Embedding Dim	Batch Size	Window Size
20	0.1	400	1000	2

Loss = 114101.66187953949

Word similarity

Nice,excellent	0.0452	Sentiment not captured

Nice,bad	0.0524	Sentiment not captured
Bad,poor	0.0202	Sentiment not captured
Nice,poor	-0.0384	Correct
Beautiful,nice	0.0217	Acceptable
Ugly,nice	-0.0591	Correct
Love,like	0.0143	Sentiment not captured
Long,longer	0.0207	Correct
Short,shorter	0.0137	Correct
Long,short	0.0548	Sentiment not captured
Longer,shorter	-0.0103	Correct

Sentiment Classifier:
Best dev accuracy: 65.95%, training accuracy: 67.18125%

***Experiment C**

Epoch	Learning Rate	Embedding Dim	Batch Size	Window Size
20	0.1	400	500	2

Loss = 223319.96677684784
Word similarity

Nice,excellent	0.1267	Correct
Nice,bad	0.0198	Correct
Bad,poor	0.0589	Correct
Nice,poor	0.0442	Sentiment not captured
Beautiful,nice	0.0670	Correct
Ugly,nice	0.0060	Correct
Love,like	-0.0176	Sentiment not captured
Long,longer	0.1891	Correct
Short,shorter	0.0624	Correct
Long,short	0.0453	Acceptable
Longer,shorter	0.0025	Correct

*Sentiment Classifier: (submitted model for prediction, myTest.csv)
Best dev accuracy: 66.225%, training accuracy: 68.1875%

Experiment D

Epoch	Learning Rate	Embedding Dim	Batch Size	Window Size
20	0.1	300	300	2

Loss = 370641.7045800686
Word similarity

Nice,excellent	0.0726	Sentiment not captured
Nice,bad	0.0997	Sentiment not captured
Bad,poor	0.1187	Correct
Nice,poor	0.0526	Sentiment not captured
Beautiful,nice	0.0932	Correct
Ugly,nice	-0.1365	Correct
Love,like	0.0939	Correct
Long,longer	0.0733	Correct
Short,shorter	-0.0083	Sentiment not captured

Long,short	0.1505	Sentiment not captured
Longer,shorter	0.1469	Sentiment not captured

Sentiment Classifier:
Best dev accuracy: 65.64%, training accuracy: 66.856%

Experiment E

Epoch	Learning Rate	Embedding Dim	Batch Size	Window Size
20	0.1	300	500	3

Loss = 223659.40666770935
Word similarity

Nice,excellent	0.1145	Correct
Nice,bad	0.0169	Correct
Bad,poor	0.0935	Correct
Nice,poor	0.0425	Correct
Beautiful,nice	0.1236	Correct
Ugly,nice	-0.0572	Correct
Love,like	0.0225	Sentiment not captured
Long,longer	0.0518	Correct
Short,shorter	0.0257	Sentiment not captured
Long,short	0.0507	Sentiment not captured
Longer,shorter	-0.0339	Correct

Sentiment Classifier:
Best dev accuracy: 66.25%, training accuracy: 65.81875%

Does the learned representation improve the result from assignment 1? If yes, why? if no, can you suggest a strategy to improve the accuracy?

No. One reason is that these are two different datasets in nature. We trained our word embedding from hotels and cars review text data. However, assignment 1 tests on twitter based text data, which the tone is more informal, and the topic is more general and less biased. A possible improvement for the analysis accuracy is to train our embedding on text data similar to twitter data, or include more different categories of topics in our training data.

Another possible improvement is to add additional layers to our model to increase the model's complexity.