This assignment will evaluate language models developed with pre-trained word vectors. Movie reviews, 500 positive and negative each, were used to train language models for predicting sentiment (positive versus negative). The assignment utilized recurrent neural networks (RNNs) in Python TensorFlow.

The movie reviews document files were read into a single list. Since the length of a review could vary, the first 20 words and the last 20 words of each review were used as word sequences. The labels for the reviews were defined as 0 for negative and 1 for positive. The data was split so there were 800 reviews in the training set and 200 reviews in the test set.

Global vectors (GloVe) provides a way to represent words as numeric vectors and was developed by individuals at Stanford University. The words are assembled into a matrix called an embedding and are pre-trained to reflect the relatedness of two words. A library called chakin was used to download the GloVe embeddings, and the GloVe.6B and GloVe.Twitter were used for the assignment. Jump-start code was utilized and modified to conduct a completely-crossed 2x2 experimental design.

The RNN utilized a basic RNN cell with 20 neurons. The output was passed through a fully connected (dense) layer which then passed the output through a softmax activation function and a cross-entropy loss was applied. Six different models were assessed with this RNN architecture. At first, the GloVe.6B embedding with 50 and 100 dimensions was compared to the GloVe.Twitter embedding with 50 and 100 dimensions. All four models used a vocabulary size of 10,000 words and were each ran three times with different random seeds as model performance would vary due to different local optima obtained with each assessment. The average test accuracy was comparable between all four models. To determine the effects of a larger vocabulary size on model performance, the vocabulary size was increased to 30,000 words and the GloVe.6B embedding with 50 and 100 dimensions was assessed for accuracy. The additional 20,000 words in the vocabulary did not have a notable effect on the average test accuracy score.

Name	Number of Dimensions	Vocab Size	Test Accuracy	Test Accuracy #2	Test Accuracy #3	Test Accuracy Average
GloVe.6B	50	10K	0.675	0.645	0.635	0.652
GloVe.6B	100	10K	0.635	0.590	0.635	0.620
GloVe.Twitter	50	10K	0.655	0.640	0.655	0.650
GloVe.Twitter	100	10K	0.640	0.680	0.640	0.653
GloVe.6B	50	30K	0.635	0.655	0.635	0.642
GloVe.6B	100	30K	0.640	0.670	0.640	0.650

Two additional models were assessed. The first model utilized a Gated Recurrent Unit (GRU) cell with the GloVe.Twitter embedding, 100 dimensions, and a vocabulary size of 100,000 words. The second model was a multilayer RNN and the code for this model was obtained and modified from Géron sample code. The multilayer RNN used the GloVe.Twitter embedding, 100 dimensions, and a vocabulary size of 10,000 words. Each model was tested three times with different random seeds to assess accuracy.

Model	Name	Number of Dimensions	Vocab Size	Test Accuracy	Test Accuracy #2	Test Accuracy #3	Test Accuracy Average
GRU cell	GloVe.Twitter	100	100K	0.740	0.73	0.755	0.742
Multilayer RNN	GloVe.Twitter	100	10K	0.555	0.55	0.555	0.553

The GRU model with a vocabulary size of 100,000 words had an average 74% accuracy, which was better than any prior model. The multilayer RNN model had an average 55% accuracy, which was the worst accuracy of any tested model.

From the benchmark study utilizing a basic RNN cell, all tested models performed similarly, even with different GloVe embeddings and increased dimensions. Increasing the vocabulary size from 10,000 to the 30,000 most popular words from the GloVe embedding did not have a notable effect. However, increasing the vocabulary size to 100,000 words and using a GRU cell provided an average 74% accuracy. Optimization of this model's hyperparameters should be pursued.