Lab Report of Exercise 2

1. Technical Description

Tabel 1 Descriptions of Main Classes and Functions

Main Classes & functions	Descriptions
Class Text2Dataset	● Read file ● preprocess the text (lower case, remove numbers and punctuations) ● generate (context, target) dataset
Class CBOW	● One hidden-layer neural network ● Turn vocabulary indices into tensor and feed into the model ● Forward learning
Function train_cbow	● Load dataset ● train model, backpropagation
Function search_words_in_different_frequencies	● Search words in different frequencies in corpus ● Return high, median, low frequency word lists
Function get_cloeset_words	● Calculate the distances between certain word and other words ● get the top n words with smallest distances

As the scifi.txt and tripadvisor_hotel_reviews.csv contain punctuations and numbers that are no good for word embedding, I removed the numbers and punctuations in the corpus. In each corpus, the same word may start with uppercase or lowercase letters, such as ‘Word’ and ‘‘word’. In order to treat them as one word, I lowered all the letters.

After I preprocess the data, I load the dataset with DataLoader(), then feed it into the defined CBOW model and train the model with different parameters. To evaluate the performance of the word embedding models, I selected some words in different frequencies and got their closest words.

2. Experiment with Iteration, Embedding Size and Window Size

I set batch size = 256, optimizer = Adam(model.parameters(), lr=0.01), then tested the model with different training epoch, embedding size and window size.

I didn't split the dataset into training and validation sets, as the cuda memory isn't enough to hold the training even with batch_size as small as 4. However I tracked the performance with different iterations.

Table 2 and Table 3 are the top 5 closest words predicted under different models on different corpus, Table 4 shows the predicted top 3 closest words of two common words in the two corpus. By analyzing the predicted closest words, I summarize as follow:

1. **Predicted closest words for noun, adjective and verb in different frequencies:**
   It's easier to find the proper closest words for adjectives and harder for verbs, for example, the predicted top 5 words for 'exceptional' are ['outstanding', 'excellent', 'amazing', 'nice', 'great'], the meaning are quite similar and the part of speech is the same.

   For high-frequency words the result is good, for example, the predicted top 5 words for 'hand' are ['shoulder', 'hands', 'finger', 'heart', 'chin'], which are all body parts as hands. But for rare words, the performance is bad. Please check the Table 2 and Table 3 for detailed predictions.

2. **Iteration**: When increasing the epoch (iteration), the 'high-frequency words' similarity worsened a bit, which can be an effect of overfitting. However, the similarity of low-frequency and rare words improved . For we have so little data about 'rare words' in the corpus, it can be hard to place them correctly in the embedding space with a few iterations, so increasing the number of iterations will improve results in "low-frequency and rare words" similarity.

3. **Embedding size**: I tried different embedding sizes, 50, 100, 30, and the best embedding size is 50.

4. **Window size**: Prediction is sensitive to window size. Compared to window_size=5, the dataset generated with window_size=2 is better in these two corpus.

5. **Bias:** Bias exists. Firstly, some words have domain bias because they have different related words in the corpus of different domains. See details in Table 4.
   Secondly, High frequency words in the corpus are common words and they generally have more similar words, for example, 'large' is a high frequency word and it has many similar words such as 'big', 'huge', 'spacious' and etc. However, the rare words themselves generally have a few similar words, so even iterated many times, the model cannot find its proper closest words because of this property but not because of data and model.

Table 2 Predicted Top 5 Closest Words on Corpus **Tripadvisor Hotel Reviews**

Trained CBOW Model	**Word: Frequency [Top 5 Closest Words]**		
	high frequency	**low frequency**	**rare**
epoch=10, window_size=2, embedding_size=50	hotel:48864 ['property', 'resort', 'place', 'desk', 'really'] good:16986 ['great', 'excellent', 'perfect', 'decent', 'better'] stay:15158 ['visit', 'return', 'staying', 'vacation', 'come']	hair:339 ['help', 'bring', 'hold', 'pack', 'washer'] exceptional:380 ['outstanding', 'excellent', 'amazing', 'nice', 'great'] imagine:355 ['understand', 'say', 'believe', 'know', 'miss']	emotions:3 ['feelings', 'veg', 'doc', 'jnr', 'shambles'] overpaying:2 ['cuba', 'powder', 'fence', 'washclothes', 'trend'] shocks:3 ['headache', 'practice', 'discounted', 'shibuya', 'blinds']
epoch=5, window_size=2, emebdding_size=50	hotel:48864 ['resort', 'property', 'option', 'place', 'hotels'] good:16986 ['great', 'excellent', 'fine', 'ok', 'better'] stay:15158 ['staying', 'visit', 'return', 'think', 'come']	hair:339 ['form', 'excited', 'want', 'using', 'proper'] exceptional:380 ['wonderful', 'incredible', 'excellent', 'fantastic', 'outstanding'] imagine:355 ['picked', 'remember', 'say', 'known', 'unless']	emotions:3 ['rm', 'picture', 'daythey', 'honeymooners', 'upgrade'] overpaying:2 ['calamari', 'familiar', 'wowed', 'gee', 'loccitane'] shocks:3 ['insist', 'consideration', 'stated', 'worn', 'add']
epoch=1, window_size=2, **emebdding_size=50**	hotel:48864 ['resort', 'property', 'place', 'hotels', 'rooms'] good:16986 ['great', 'excellent', 'decent', 'fantastic', 'better'] stay:15158 ['really', 'staying', 'experience', 'visit', 'vacation']	hair:339 ['washer', 'checkout', 'including', 'no', 'asking'] exceptional:380 ['wonderful', 'outstanding', 'offered', 'fantastic', 'especially'] imagine:355 ['believe', 'think', 'say', 'means', 'really']	emotions:3 ['staff', 'photographers', 'pp', 'man', 'container'] overpaying:2 ['rincon', 'seedier', 'bravo', 'wo', 'someday'] shocks:3 ['certain', 'american', 'unusual', 'giving', 'common']

epoch=1, window_size=2, **emebdding_size=100**	hotel:48864 ['property', 'resort', 'place', 'quite', 'really'] good:16986 ['great', 'excellent', 'fantastic', 'better', 'nice'] stay:15158 ['staying', 'visit', 'hotel', 'return', 'experience']	hair:339 ['walk', 'lovely', 'loved', 'fantastic', 'rest'] exceptional:380 ['fantastic', 'amazing', 'wonderful', 'better', 'excellent'] imagine:355 ['say', 'think', 'believe', 'pool', 'complain']	emotions:3 ['drainage', 'jug', 'signature', 'bagel', 'shower'] overpaying:2 ['crepe', 'maintenance', 'frank', 'invention', 'waves'] shocks:3 ['agree', 'writing', 'mention', 'truly', 'use']
epoch=1 , window_size=2, **embedding_size = 30**	hotel:48864 ['property', 'hotels', 'point', 'equally', 'westin'] good:16986 ['great', 'decent', 'better', 'excellent', 'fine'] stay:15158 ['staying', 'experience', 'visit', 'vacation', 'come']	hair:339 ['blow', 'awkward', 'having', 'completely', 'contains'] exceptional:380 ['notch', 'excellent', 'outstanding', 'class', 'terrible'] imagine:355 ['believe', 'say', 'possibly', 'wait', 'remember']	emotions:3 ['toll', 'passengers', 'ourroom', 'broadband', 'freeway'] overpaying:2 ['villages', 'greatoorfurnishings', 'dave', 'refills', 'interestthere'] shocks:3 ['lucie', 'shows', 'perserve', 'japenese', 'hottest']
epoch=1 , **window_size=5** , embedding_size = 50	hotel:48864 ['place', 'time', 'just', 'room', 'quite'] good:16986 ['fine', 'great', 'excellent', 'nice', 'best'] stay:15158 ['vacation', 'return', 'experience', 'hotel', 'just']	hair:339 ['ceiling', 'basic', 'brought', 'missing', 'bargain'] exceptional:380 ['fantastic', 'excellent', 'wonderful', 'outstanding', 'perfect'] imagine:355 ['wrong', 'say', 'overall', 'said', 'really']	emotions:3 ['thomson', 'reports', 'comments', 'aerobus', 'reviews'] overpaying:2 ['beds', 'soda', 'burger', 'sinks', 'antiques'] shocks:3 ['malfunctioned', 'contains', 'nationalities', 'ruined', 'recovered']

Note: The red words are the ones that I marked as proper predicted closest words.

From the above table, we can see that the predictions for low frequency and rare words are better with epoch=10 than with epoch=1; but for high frequency words, the predictions are better when epoch=1. Model with Embedding size=50 is better than embedding size=30 or 100. Dataset generated with window_size=2 has better performance than with window_size=5.

As the data scifi has much larger corpus than hotel reviews and words appear more often, I set epoch=1, window_size=2 to train models and only compare embedding size on this corpus.

Table 3 Predicted Top 5 Closest words on Corpus **Scifi**

Trained CBOW Model	**word: frequency [Top 5 Closest Words]**		
	high frequency	**low frequency**	**rare**
epoch=1, window_size=2, **embedding_size =100**	hand:10996 ['face', 'hands', 'arm', 'mind', 'chair'] good:15435 ['real', 'just', 'great', '', 'that'] know:28539 ['remember', 'say', 'think', 'believe', 'thought']	cup:596 ['silence', 'line', 'rest', 'water', 'rush'] amazing:528 ['though', 'confused', 'quiet', 'if', 'passing'] aid:679 ['out', 'promise', 'knowledge', 'back', 'word']	timetables:1 ['legitimate', 'defeated', 'chosen', 'starved', 'monitored'] thankless:4 ['talisman', 'suburban', 'long', 'rush', 'deep'] trys:2 ['another', 'morgan', 'kindly', 'yesterday', 'everything']
epoch=1, window_size=2, **embedding_size =50**	hand:10996 ['shoulder', 'hands', 'finger', 'heart', 'chin'] good:15435 ['fine', 'bad', 'great', 'nice', 'small'] know:28539 ['mean', 'understand', 'think', 'say', 'remember']	cup:596 ['pack', 'stream', 'bag', 'bottle', 'flare'] amazing:528 ['impossible', 'possible', 'imperfect', 'rapid', 'abrupt'] aid:679 ['violence', 'victory', 'satisfaction', 'fear', 'fugue']	timetables:1 ['ally', 'absurdly', 'intending', 'rousseau', 'tickets'] thankless:4 ['ecological', 'quantum', 'highpower', 'screwing', 'antiaircraft'] trys:2 ['gibson', 'kaifri', 'everything', 'vix', 'resigned']

For this corpus, the embedding size = 50 is also better than embedding_size=100.

Table 4 Predicted Top 3 Closest Words for **Common Words** in the Two Corpus

Models trained on the two corpus with batch_size=256, epoch=1, embedding size=50, window size=2
optimizer = optim.Adam(model.parameters(), lr=0.01)

Common words	**Predicted on scifi**	**Predicted on tripadvisor hotel reviews**
day	['time', 'night', 'world']	['everyday', 'week', 'days']
place	['world', 'country', 'town']	['hotel', 'resort', 'overall']
good	['fine', 'bad', 'great']	['great', 'fine', 'excellent']

As the result shows, the prediction has domain bias. For example, for the common word ‘place’, the top 3 predicted closest words on corpus scifi are: world, country and town; however on corpus tripadvisor hotel reviews, the proper predicted words are: hotel and resort.