CS310 Natural Language Processing - Assignment 2: Word2vec Implementation

Total points: 50

Task: Train a word2vec model using the skip-gram architecture and negative sampling.

- The corpus data being trained on is the full text of 《论语》.
- Use the Lab 4 content to help you.

Submit:

- The modified notebook file `A2_w2v.ipynb`
- A zipped file containing all resulted word embeddings (.txt format)
- Any dependent Python files.
- Write up the results for the following requirement 3 and 5 in a Word/PDF document.

Requirements:

1) (10 points) Implement the data loading and processing pipeline. You should re-use and augment the code for `generate_data` and `batchify` functions.

2) (15 points) Implement the `SkipGram` class. The key is to implement the computation for `loss` in `forward` function. Make sure the inputs to this function are tensors in correct dimensions.

3) (10 points) Implement the `train` function that runs correctly.

   a) Print the loss every few intervals (determine the number by your observation). Include a screenshot of loss change in your write-up.

   b) Determine the training epochs needed by observing when the loss stops decreasing significantly.

4) (10 points) Run training with different hyper-parameters; save the embedding results.

   a) Train with `emb_size` = 50, 100, respectively

   b) Train with negative sample number `k` = 5,10,15, respectively

   c) Train with `window_size` = 1, 3, 5, respectively

   Therefore, there are in total $2 \times 3 \times 3 = 18$ experiment groups, that is, 18 embedding files need be submitted.

4) (5 points) Plot and compare the embeddings with LSA ones.

   a) Use Truncated SVD to reduce the dimension of embeddings from the target words provided (`['学', '习', '曰', '子', '人', '仁']`). Plot all of the 18 embedding results. You may also use the words that you find interesting instead.

   b) Compare your favorite embedding plot with the one we obtained from the LSA Lab. Briefly describe the difference in your write-up