Final Assignment

Some background

Word2vec is a technique for natural language processing published in 2013.
The word2vec algorithm uses a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence. As the name implies, word2vec represents each distinct word with a vector. The vectors are chosen carefully such that the cosine similarity (equivalent to distance for unit length vectors) between the vectors indicates the level of semantic similarity between the words represented by those vectors.

The original paper
https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

It is recommended to use blogs and other web resources to understand the model
https://www.ccs.neu.edu/home/vip/teach/DMcourse/4_TF_supervised/notes_slides/Word2Vec%20Tutorial%20-%20The%20Skip-Gram%20Model%20%C2%B7%20Chris%20McCormick.pdf
Which is also attached to the assignment

The Task

Attached are some hotel reviews corpus (zipped file). Each line in the file is a review.

1. Implement and train **word2vec skipgram model** using the functional API of tf.keras.
2. Implement a function that given a word finds the closest embedded (with respect to cosine similarity) words to it along with their respective cosine similarity score. Below k is the number of closest words to return

   ```
   def find_most_similar(word, k=10):
       …
   ```

3. Use a 2d or 3d projection (dimensionality reduction) and plot some of the words you find interesting.
4. **(bonus)** Use k-means to find some interesting clusters of words (embeddings)

Notes:

- Remember that in order to use words (categorical variable) you need to map them to indices (integers). It is useful to build also the inverse mapping: index -> word
- You can decrease the size of the vocabulary by removing rare words (or very frequent no interesting words - aka “stopwords”). When doing so, remember to replace all the removed words with a special token so as not to affect the windowing.
- When building the model, you are required to choose hyper parameters like window size and embedding size.

Hints: (not mandatory to use them)

You can look at
https://www.tensorflow.org/tutorials/text/word2vec
For implementation using the subclass API

For generating skipgrams pairs check the function
tf.keras.preprocessing.sequence.skipgrams

For negative sampling check the function
tf.keras.preprocessing.sequence.make_sampling_table