Homework 3

For this assignment, you will experiment with word2vec, an algorithm for training word embeddings. I suggest that you use gensim (https://radimrehurek.com/gensim/) for these experiments, but other implementations, such as Mikolov's original code (https://code.google.com/archive/p/word2vec/) are acceptable too. We will run a few basic tests on the model that you will train but you may need the word embeddings that you will generate for the next homework and/or your project.

If you decide to use gensim for your experiments, begin by studying this tutorial:

https://radimrehurek.com/gensim/auto_examples/tutorials/run_word2vec.html

When you have a good understand of how to use gensim (and it's pretty easy!), please complete the following steps:

1. Train a word2vec model on a corpus of your choice. Here are a few corpora ideas, but please feel free to use something else (and please share on Teams if you find something interesting!):

   - WikiText – added new (04/21/2022) (https://blog.salesforceairesearch.com/the-wikitext-long-term-dependency-language-modeling-dataset/)
   - https://anc.org
   - https://www.linguistics.ucsb.edu/research/santa-barbara-corpus
   - http://mattmahoney.net/dc/textdata.html

   If these corpora are too large for your hardware to handle, you can use Azure for training. Alternatively, feel free to use a 10-25% chunk of one of these datasets (but the quality of your word embeddings may suffer).

   When the training completes, validate your model by running a few similarity queries. Include a few examples of similar words and their similarity scores in your report. (25 points)

2. Load a large pre-trained model (e.g. you can use the Google News model that's mentioned in the tutorial) and test it qualitatively. One idea would be to identify a few clusters of similar words that belong to different categories (e.g. people, places, colors, etc.). Explain why word2vec considers these words similar. (25 points)

3. Evaluate the large model on the WordSim-353 dataset, which you can download here: http://alfonseca.org/eng/research/wordsim353.html Please use the gold standard similarity file (wordsim_similarity_goldstandard.txt). The evaluation should be done by measuring the Spearman's rank correlation coefficient:

   http://en.wikipedia.org/wiki/Spearman_rank_correlation

   There are many implementations available such as:

   https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.spearmanr.html

Once you have a result, please post it on Teams for other people to compare their results to yours. (25 points)

4. Propose several examples of word analogies (e.g. ‘king’ - ‘man’ + ‘woman’ = ‘queen’) and test them using the large model. (25 points)