# 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

## The Task

Attached are some hotel reviews corpus. It is saved in parquet format and should be read using Pandas. Each row in the dataframe  is a review composed of a title and a text field. This assignment is also accompanied by a notebook that provides some instructions and structure. Use it.

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:

- Output cells should not be too big. DO NOT DUMP A LOT OF DATA IN THE OUTPUT. NOTEBOOKS THAT WON'T FOLLOW THIS INSTRUCTION WILL NOT BE CHECKED

AND GRADE WILL BE SET TO ZERO.
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.