Goal: Measure the similarity between words, and describe your findings with respect to the similarity scores

A. Experimental Setup: Model and Data

We need
-- A model that represents the meaning of words by vectors
-- A list of word pairs

Steps to take:
1. Run exercise1_advanced.py to make sure all the required python modules and data are installed and get familiar with the script.
If you run into errors, follow the instructions printed in the terminal and within exercise1_advanced.py.

2. Get a pre-trained semantic model that learnt word embeddings on the Google news corpus using a prediction-based method.
Detail to the model: It is a skip-gram model (word2vec),
Mikolov, T., Yih, W., & Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. https://www.aclweb.org/anthology/N13-1090

The script downloads a pruned version of it.

Note:
You can get more background to the whole exercise also in this blog post:
https://www.kaggle.com/alvations/word2vec-embedding-using-gensim-and-nltk

3. (Optional) You can also download the full version later (see also the code):
File name: GoogleNews-vectors-negative300.bin.gz
Source: https://code.google.com/p/word2vec/

B. Exercise in class + at home, results to be submitted
Data analysis and presentation of the method and the findings next week in class.

Notes:

- All the functions you need in order to solve the tasks are documented on https://www.pydoc.io/pypi/gensim-3.2.0/autoapi/models/keyedvectors/index.html
  And
  https://radimrehurek.com/gensim/models/keyedvectors.html
- You can also use the script ex1_word-similarity.py to find out which statements to use.

Tasks
For each of the three words "apple", "sweet", and "allow", calculate a)-d).

Take notes of your results and your observations, such that you can present them next week.

a) 20 nearest neighbors of the words in a space built on the google news-gram corpus, using the word2vec semantic model.

b) 10 nearest neighbors for two-word phrases that you associate to different senses of the word (e.g., for "mouse", "use mouse" vs. "furry mouse").

c) Cosine similarity between the word and:
- one (near-)synonym,
- one topically-related word,
- one hypernym,
- one antonym (if available)

You can get the data from WordNet (http://wordnetweb.princeton.edu/perl/webwn) or any other dictionary, or based on your intuition. See Chapter 19 of the 2nd edition of SLP for definitions of synonym, hypernym, etc. (available on the course's Aula Global).