**Ingredients text analysis with deep learning using Word2Vec**

Siying Cen

In this assignment, I apply distributed word vectors (word2vec) to a recipe dataset.

(<https://www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions>)

This dataset contains more than 30,000 words of ingredients in the recipes and my goal is to find the words similarity.

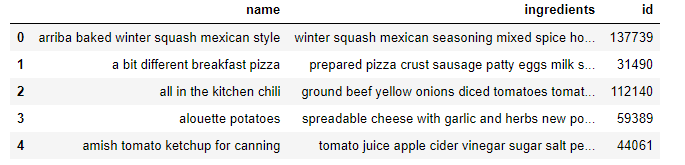
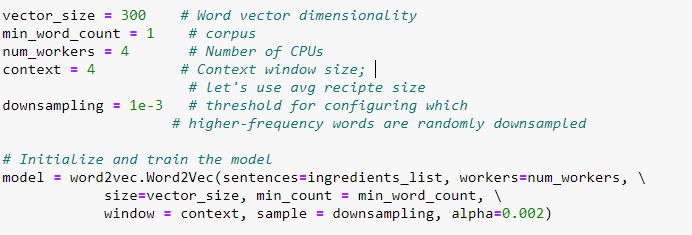


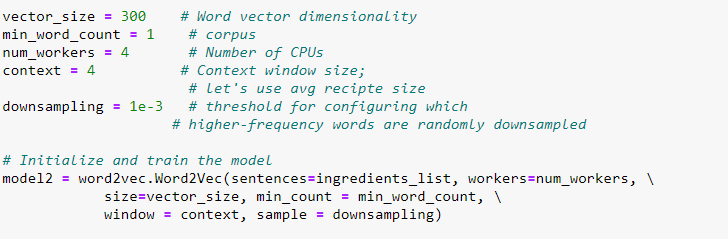
Figure 1. Screenshot of food and recipes dataset.

The algorithm I used is word2vec in Gensim (URL of this module is <https://radimrehurek.com/gensim/models/word2vec.html>). The word2vec algorithms include skip-gram and CBOW models, using recurrent neural networks to learn and negative sampling.

The coding environment is Jupyter notebook. After importing dataset, first I find the 5 most common and 5 least common ingredients used in the recipes.

I built 3 word2vec models for training and adjust some of their parameters: the vector space size used in all models is 300 and down sampling is 1e-3; in model 1 and 2, the minimum word count is 1 while in model3, the minimum word count is 4, which means that the training ignore all words with total collected count lower than 4; in model 2 and 3, the learning rate is default value alpha=0.025 while in model1, I changed the learning rate to 0.002.





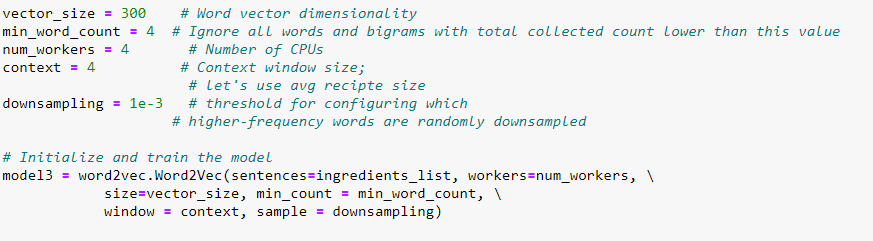


Figure 2. Screenshots of model 1, model 2 and model 3.

After building these models, I choose 10 same words (“lemon”, “lettuce”, “wine”, “flower”, “flour”, “crab”, “oil”, “pasta”, “potato”, “chicken”) to test every model. I used “model.wv.most\_similar(‘words’)” command to return the most similar words of these testing words with cosine similarity values. The results were shown in the jupyter notebook. For model1, the best performance is 0.99, the average performance is 0.85 and the worst performance is 0.54. For model2, the best performance is 0.93, the average performance is 0.65 and the worst performance is 0.34. For model 3, the best performance is 0.91, the average performance is 0.65 and the worst performance is 0.33.

Model 1 with the lowest learning rate shows the best performance, while the minimum word count does not affect a lot. Therefore, learning rate is an important parameter for word2vec model training.