## HU Extension Assignment 08 E63 Big Data Analytics

### Handed out: 10/20/2017 Due by 4:00 PM EST on Saturday, 10/28/2017

If you are familiar with NLP API-s in languages other than Python or Python NLP API-s other than NLTK please be free to solve these problems using technology of your choice.

**Problem 1.** Use the text of the Universal Declaration of Human Rights (UDHR). Create a table for 5 languages in which you will collect statistics about the languages used. Place in that table the number of words in each language in UDHR, number of unique words, average length of words, number of sentences contained in UDHR and average number of words per sentence. Create a distribution of sentence lengths for each language. Plot those (non-cumulative) distributions on one diagram.

**(25%)**

**For problem 1 to 3, please refer to HW8\_HV28Oct17.ipynb**

**Problem 2**. Identify 10 most frequently used words longer than 7 characters in the entire corpus of Inaugural Addresses. Do not identify 10 words for every speech but rather 10 words for the entire corpus. Which among those words has the largest number of synonyms? List all synonyms for those 10 words. Which one of those 10 words has the largest number of hyponyms? List all hyponyms of those 10 most frequently used “long” words. The purpose of this problem is to familiarize you with WordNet and concepts of synonyms and hyponyms.

**(25%)**

**For problem 1 to 3, please refer to HW8\_HV28Oct17.ipynb**

Your literature for Problems 1 and 2 are chapters 1 and 2 of Natural Language Processing with Python book by Steven Bird et al.

**Problem 3.** Create your own grammar for the following sentence:

“Describe every step of your work and present all intermediate and final results in a Word document”.

**(10%)**

Your literature for Problem 3 is chapter 8 of Natural Language Processing with Python book by Steven Bird et al.

**For problem 1 to 3, please refer to HW8\_HV28Oct17.ipynb**

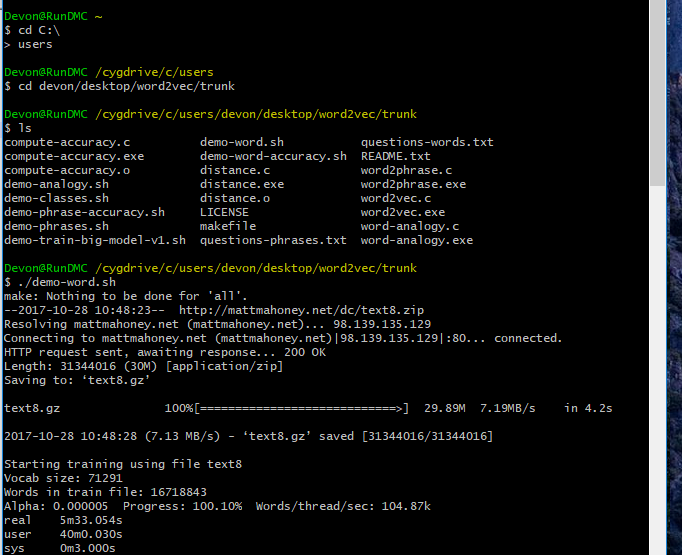
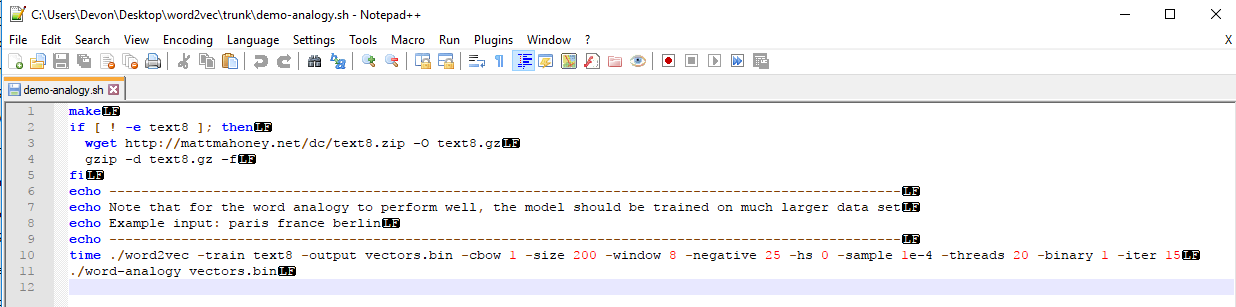
**Problem 4.** Install and compile Word2Vec C executables. Train CBOW model and create 200 dimensional embedding of Word Vectors. Demonstrate that you could run analogical reasoning when searching for country’s favorite food starting with japan and sushi. Note that words might have to be in lower case. Find favorite food for 5 different countries. Report improbable results as well as good results. Use scripts provided with original Google C code.

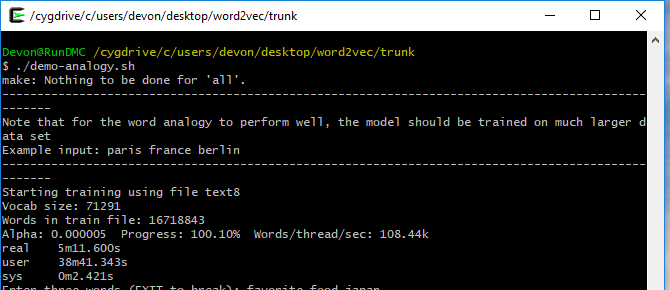
(**20%)**

***Im using my boyfriend’s desktop instead of my laptop, his Desktop has 4 processors and 16GB of RAM. My laptop is very, very slow. His name is Devon, so you will see users/devon/…..***

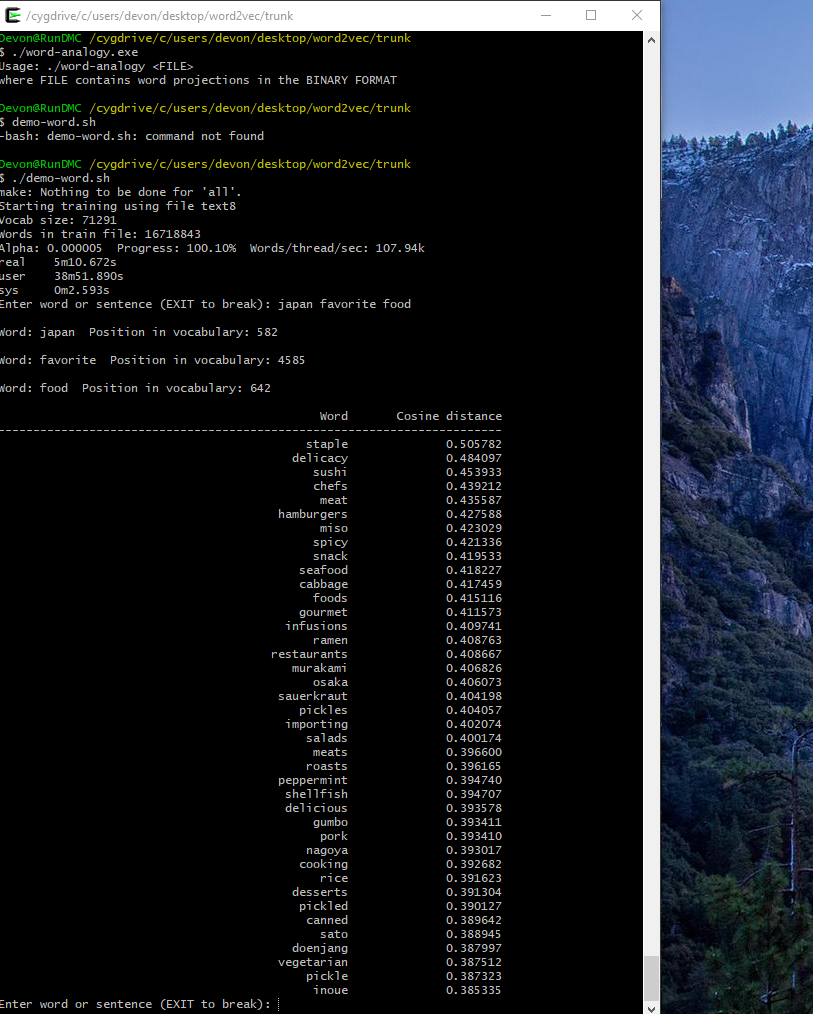
I unzipped source-archive file, and gcc-7.2.0.tar. I linked gcc with cygwyn, then open demo-word.sh

# I tried word-analogy.sh but the results looked nothing to do with sushi, only Japan towns were listed. So I used demo-word.sh and it works!

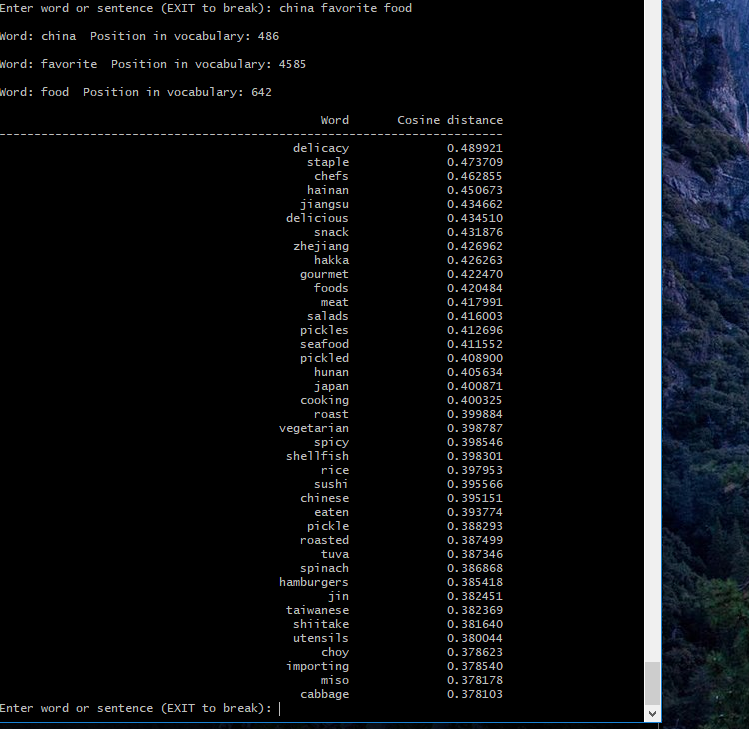




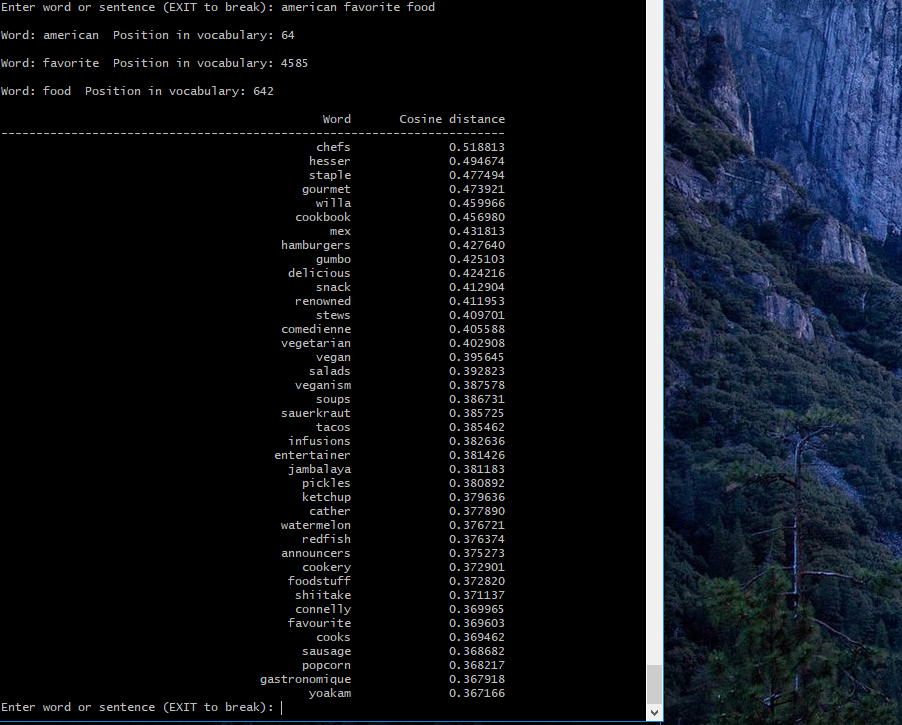
**#1\_Japan sushi:**



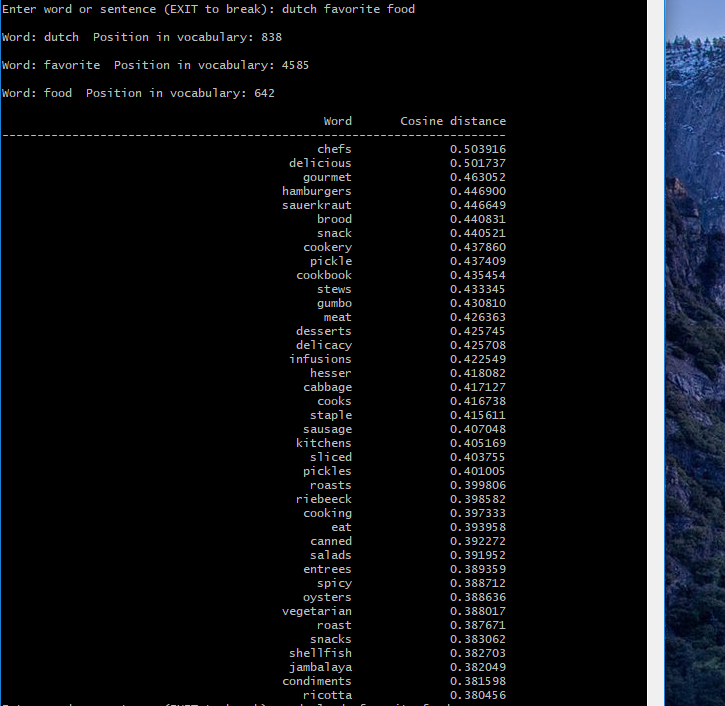
**#2\_China favorite food:**



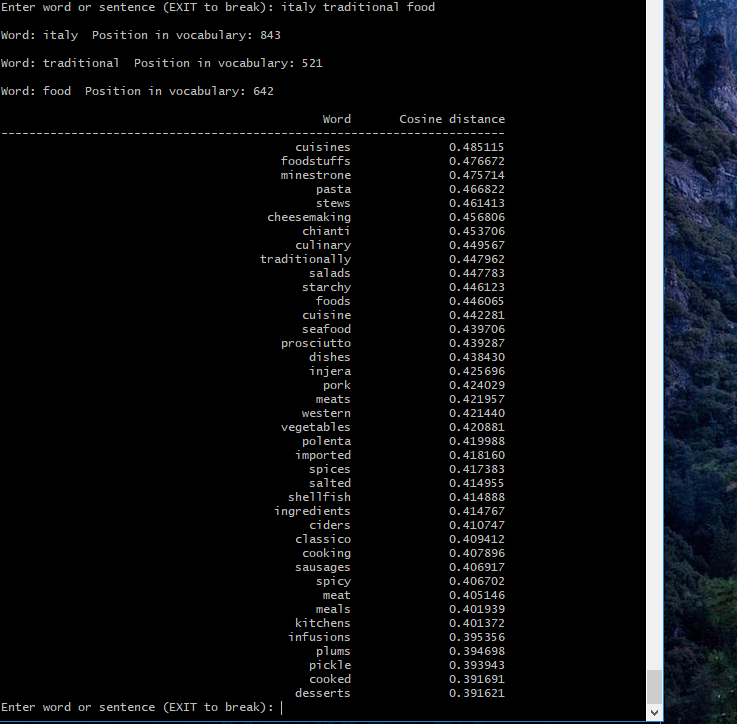
**#3\_America favorite food:**



**#4\_Dutch favorite food**



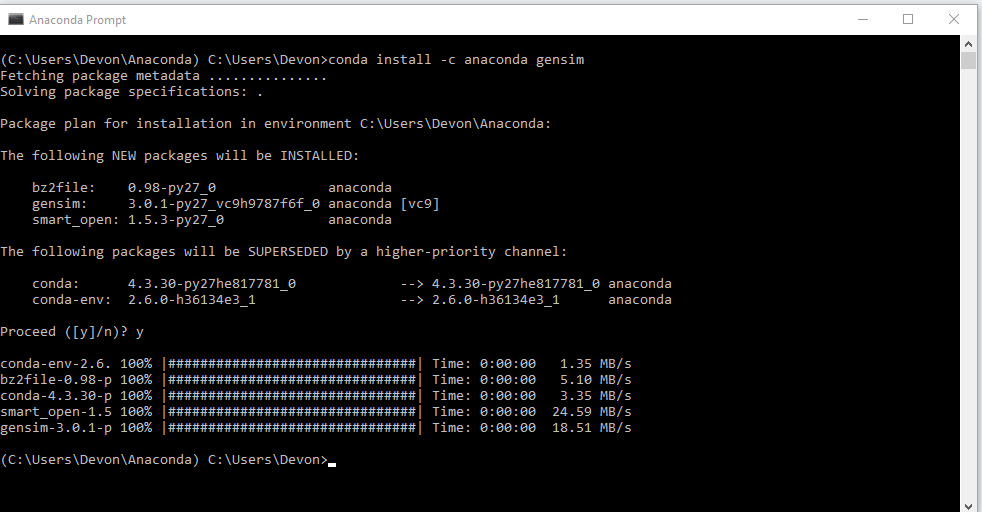
**#5\_Italy traditional favorite food**



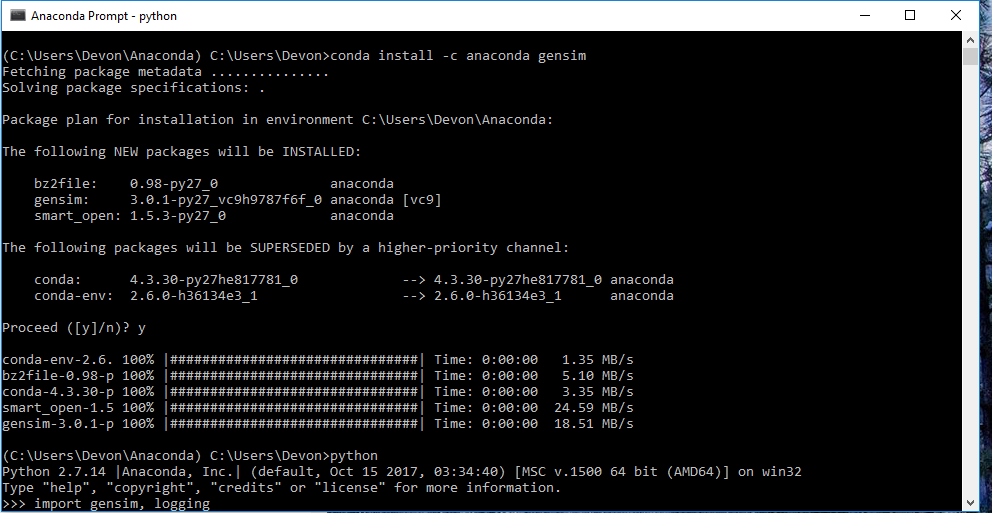
**Problem 5.** Install and run Genism Python Word2Vec API. Find the most probable words you will obtain when you start with an emperor add a woman and subtract a man. Use this tutorial as a guide <https://rare-technologies.com/word2vec-tutorial/>

**(20%)**

Install Gensim in Anaconda



I copied vectors.bin from C:\Users\Devon\Desktop\word2vec\trunk to C:\Users which is where Anaconda folder is located. Then, I open Anaconda Promt-python



#start training for emperor + woman – man = empress

