Northeastern University, Khoury College of Computer Science

CS 6120 Natural Language Processing — Lab 7

February 20, 2025 (Week 7)

Word2Vec

In this lab, we're going to explore Mikolov's original `word2vec` code, written in `C`, which was originally released here after publication. Many of the links don't work anymore, including the link to the code on subversion, so you will need to clone it from Github. The code itself is robust and should run out of the box.

1 Installing Software

The C code was developed and works on Linux / Bash machines. It can work on Mac and Windows, but might need some tweaks. (For example, I had to change `fgetc_unlocked` to `getc_unlocked` on my Macbook Pro.) If you don't have a Linux / Bash machine (and I don't expect that anyone does), then you can either create a VM on GCP or use any Docker container. (Most Docker containers are in Linux, and the Docker image you created in Lab 2 works just fine.)

Go ahead and clone `word2vec` from Mikolov's repository, with the following line:

```
$> git clone https://github.com/tmikolov/word2vec
```

You will need to install C and C++. Check out some resources for this here.

2 Run Word to Vec

Try out the demonstration with `demo-word.sh`. This will download a file called `text-8k`, train your neural network vectors, writes them out to file, and explores the vectors through an interactive command line. Depending on your machine. Running *could* take a little while, depending on your machine. Notice in the shell script that it is default to *twenty threads*.

```
$> ./word2vec -train text8 -output vectors.bin -cbow 1 -size 200 -window 8 \
   -negative 25 -hs 0 -sample 1e-4 -threads 20 -binary 1 -iter 15
```

However, you don't need to train to completion if you don't like. Just take a screenshot of it training. If you aren't training to completion, you can download the vectors from our website. You can do that with the following command:

```
$> wget https://course.ccs.neu.edu/cs6120s25/data/wikipedia/vectors-words.bin
```

Using either `vectors-words.bin` or the vectors that you have trained, choose five interesting words and print out the nearest vectors to these words with their distances. The code to do this is entirely located in this repository.

3 Loading Vectors In with Gensim

It is often more useful to load the data into Python and manipulate the vectors, visualize, and calculate distances interactively. To do so, load the data into Python with the following code.

```
from gensim.models import KeyedVectors
model = KeyedVectors.load_word2vec_format('vectors.bin', binary=True)
```

Gensim is a package with considerable NLP capabilities. You will need to install it with `pip install gensim`. Then, after loading the file by reading the `vectors.bin` file, you can access the vectors with

```
vector = model['word']
```

Go ahead and play around with the vectors in Python, but take a screenshot of a vector that you've loaded into Python.

4 Submit Your Screenshot

Go to Gradescope and submit the screenshots of (1) word2vec training, (2) the nearest neighbors and their distances to 5 words of your choice, and (3) iPython, Python, or Jupyter Notebook of a word vector.