CIS8045 Assignment 3 Alice's Adventures in Wonderland

** For most parts of this assignment, you should be able to use some parts from the demo code in class.

Part I. Text Import

Download the **'gutenberg'** corpus through NLTK. Import NLTK using the following command:

```
import nltk
```

Then, use the **nltk.download()** command to open up an app, that is, the **NLTK Downloader** interface (shown in the following screenshot):

```
nltk.download()
```

The app has multiple tabs. Click the **Corpora** tab, scroll down until you reach **gutenberg**. If the status is **not installed**, go ahead and click the **Download** button in the lower-left corner. That should install the **gutenberg** corpus. Read in the text and store it in a variable:

```
alice_raw = nltk.corpus.gutenberg.raw('carroll-alice.txt')
```

Part II Text Preprocessing

Work with the text for Alice in Wonderland that we stored in the **alice_raw** variable. The following are the steps you need to perform:

1. Continuing in the same Notebook, use the raw text in the **'alice_raw'** variable. Change the raw text to lowercase.
2. Tokenize the sentences.
3. Import punctuation from the **string** module and the stop words from NLTK.
4. Create a variable holding the contextual stop words, that is, **--** and **said**.
5. Create a master list for stop words to remove that contain terms from punctuation, NLTK stop words and contextual stop words.
6. Define a function to drop these tokens from any input sentence (tokenized).
7. Use the **PorterStemmer** algorithm from NLTK to perform stemming on the result.
8. Print out the first five sentences from the result.

Part III Text Representation

Continue using the same Notebook. Work on the result of the stop word removal step we got (let's say it is stored in a variable called **alice_words_nostop**). Print the first three sentences from the result.

1. Import **word2vec** from Gensim and train your word embeddings with default parameters.
2. Find the terms most similar to **rabbit**.
3. Using a window size 2, retrain the word vectors.
4. Find the terms most similar to **rabbit**.
5. Retrain the word vectors using the Skip-gram method with a window size of **5**.
6. Find the terms most similar to **rabbit**.
7. Find the representation for the phrase **white rabbit** by averaging the vectors for **white** and **rabbit**.
8. Find the representation for **mad hatter** by averaging the vectors for **mad** and **hatter**.
9. Find the cosine similarity between these two phrases.
10. Load pre-trained GloVe embeddings of size 100D.
11. Find representations for **white rabbit** and **mad hatter**.
12. Find the cosine similarity between the two phrases. Has the cosine similarity changed?

As a result of Part II, we will have our own word vectors that have been trained on "Alice's Adventures in Wonderland" and have representation for the terms available in the text.