W4 Assignment – Corpus Linguistics 2

Computational Linguistics

John Gamboa

April 22, 2022

1 The Brown Corpus

This week I'd like to try something new. I want to have you explore a corpus on your own. I have the impression that watching vides is good to get an idea about the topics, but you'll only really learn if you actually write the Python code on your own.

The following questions will try to guide you in your exploration of the Brown Corpus. It is a general corpus of the English language, the first one with 1M words, and, as we saw very en passant in the last class, it can be accessed from the `nltk.corpus.brown` subpackage. I'll leave a lot of information just open in the questions. The hope is that it will force you to explore a little more. If you have doubts, you are welcome to use the forum.

Before you start

Open a new Jupyter Notebook and run the line:

```
import nltk
```

You'll access mostly functions that have to do with the `nltk.corpus.gutenberg` package. As the questions go, I'll ask you to run certain functions.

1.1 Files in the corpus

If you read Gries (2009), you'll remember that corpora are often a collection of text files. You might not have realized, but this is why the `nltk.corpus.gutenberg` package comes with the function `.fileids()`. In the gutenberg package, the file ids had an obvious meaning: each of them was a book; in the case of the Brown Corpus, the files do not have obviously identifiable names.
If you run the function

```
nltk.corpus.brown.fileids()
```

you'll see the names of the files. I'd like you to find out how many files are there in the corpus. Select the correct option:

○ 800 ○ 700 ○ 500 ○ 844 ○ 685 ○ 1000 ○ 100 ○ 399

1.2 Making sense of the files

There is a better way to access the files in the corpus: through its categories. The nltk provides the function

```
nltk.corpus.brown.categories()
```

for that. How many categories are in the corpus?

`answer:` ________________________________

1.3 Accessing the corpus words

Remember that the gutenberg package had a function

```
nltk.corpus.gutenberg.words()
```

to which you could optionally pass an argument with the name of the file. This function also exists in the Brown corpus:

```
nltk.corpus.brown.words()
```

Optionally, you can pass the name of a category:

```
nltk.corpus.brown.words(categories='mystery')
```

How many unique words do you have in this "mystery" category? (tip: remember the previous classes)

answer: ______________________________

1.4 Checking conditional probabilities

In this week's class we've learnt about the `nltk.ConditionalFreqDist()` objects. Create a ConditionalFreqDist object with the data of the "mystery" and "news" genres (passing the genre as the "condition").

Now... the following is a list of modal verbs in the English language:

```
modals = ['shall', 'should', 'can', 'could', 'ought', 'may',
          'might', 'must', 'will', 'would']
```

There is one verb that is used very differently in these two genres. Using the object you created, find out which one that is.

answer: ______________________________

1.5 Accessing sentences

Similar to the gutenberg package, the NLTK package containing the Brown Corpus also has a function for accessing sentences:

```
nltk.corpus.brown.sents()
```

Again, you can optionally pass a category as a parameter to the function:

```
nltk.corpus.brown.sents(categories='mystery')
```

Can you find out the average sentence length of the “news” genre?

answer: ________________________________