CIS 4930 NLP // HW #8 // Spring 2018

Date Assigned: March 20, 2018
Date Due: March 23, 2018

Submission Format

You will submit a soft copy of your solution using e-Learning (http://elearning.ufl.edu) by the end of the day (23:59 / 11:59 PM) on the assigned date (March 23). Submit one file, **hw7.py**.

Assignment

At the top of every solution file you submit this semester include: your name, section number, the assignment number, and the date due. Complete the class POS_Tag_Data by implementing the necessary code for __init__, all_tag_inds, and all_word_inds. A class template and sample test cases are given below.

Exercises

__init__: construct the object and initialize its properties.

- *tagged* – the tuples of the word/tag pairs from the corpus provided. Make all *words* are *lower-case* but leave the tags unchanged (i.e. leave them upper-case if they are). Remove the punctuation from this list if **punctuation=False**. Use the *universal* tagset if **universal=True**.
- *tags* – a list of just the tags. Note the index positions of *tags* will correspond to those of *tagged* and *words*.
- *words* – a list of just the words (lower-case). Note the index positions of *words* will correspond to those of *tagged* and *tags*.

all_tag_inds: searches *tags* for the **tag** provided and returns a list of the index positions where **tag** is found.

all_word_inds: searches *words* for the **word** provided and returns a list of the index positions where **word** is found.

Template

```
import nltk, re
from nltk.corpus import brown, treebank

class POS_Tag_Data :

    # given a corpus,
    # place the tuples of word/tag pairs into tagged
    # make the words in tagged / words lists lower case
    # use the punctuation selection provided
    def __init__( self, corpus, punctuation=False, universal=True ) :
        tagged = []
        tags = []
```

```
        words = []

        # when punctuation is False, take it out
        # when universal is True, use the universal tagset

        self.tagged = tagged
        self.tags = tags
        self.words = words

    # find all index positions of the tag provided
    def all_tag_inds( self, tag ) :
        inds = []

        # find the inds

        return inds

    # find all index positions of the word provided
    def all_word_inds( self, word ) :
        inds = []

        # find the inds

        return inds
```

Test Cases

```
# sample test cases to consider
ptd_brown = POS_Tag_Data( brown )
noun_inds = ptd_brown.all_tag_inds( 'NOUN' )
work_inds = ptd_brown.all_word_inds( 'work' )

ptd_treebank = POS_Tag_Data( treebank, universal=False )
nnp_inds = ptd.all_tag_inds( 'NNP' )
word_inds = ptd.all_word_inds( 'work' )
```