**Coding Assignments 3**

Due 2/17 11:59 pm

**Tasks**:

* Download all news articles from [http://web.archive.org/web/20200613003232if\_/http://feeds.reuters.com/Reuters/worldNews](http://web.archive.org/web/20200613003232if_/http:/feeds.reuters.com/Reuters/worldNews). Use BeautifulSoup to extract, for each news article, the title/headline, the publication time, and the text. Store the extracted information into a Pandas DataFrame so that each news article is an entry and the columns include Title, Time, Content.
* On the above dataset, cleanup the data using the clean and normalize functions and generate, for each article, lemmas, nounds, noun-phrases, and entity pairs. Combine them into a dictionary in the following form and save it to a file in a format of your choice: txt, json, csv, pkl:

{"Lemma": [list of lemmas], "Noun": [list of nouns], "Noun Phrase": [list of noun phrases], "Entity Pairs": [list of entity pairs]}

**Hints:**

* Watch grader's Zoom-CA03 videos to learn a snippet for downloading html files, saving them as html files on your hard drive, and processing them using BeautifulSoup.

import html

feed\_reuters = feedparser.parse(  
    '<http://web.archive.org/web/20200613003232if_/http>://feeds.reuters.com/Reuters/worldNews')  
CA\_DIR = "CA03" ### you need to create directory CA03 in your code directory for downloaded files  
for url in feed\_reuters.entries[:]:  
    file\_name = url.id.split("/")[-1] + ".html"  
    html = requests.get(url.id)  
    path = os.path.join(CA\_DIR, file\_name)  
    with open(path, "w+") as f: ### w+ opens a file for both writing and reading  
        f.write(html.text)   
        f.close()  
    soup = BeautifulSoup(html.text, 'html.parser') ### read in an HTML file and work on it

* For details how to save a dictionary in one of the aforementioned formats you may consult <https://pythonspot.com/save-a-dictionary-to-a-file/>.
* Use display\_nlp by removing irrelevant parts to extract what is needed.
* Modify extract\_noun\_phrases to extract nouns.
* When running the clean function, you may experience an Attribute Error on unescapte(). Modify it by adding an exception handling as follows:

import html

### Standard cleaning function  
def clean(text):  
    # convert html escapes like & to characters.  
    try:   
        text = html.unescape(text)   
    except AttributeError:  
        print("Attribute error: ignored")  
    # tags like <tab>  
    text = re.sub(r'<[^<>]\*>', ' ', text)  
    # markdown URLs like [Some text](https://....)  
    text = re.sub(r'\[([^\[\]]\*)\]\([^\(\)]\*\)', r'\1', text)  
    # text or code in brackets like [0]  
    text = re.sub(r'\[[^\[\]]\*\]', ' ', text)  
    # standalone sequences of specials, matches &# but not #cool  
    text = re.sub(r'(?:^|\s)[&#<>{}\[\]+|\\:-]{1,}(?:\s|$)', ' ', text)  
    # standalone sequences of hyphens like --- or ==  
    text = re.sub(r'(?:^|\s)[\-=\+]{2,}(?:\s|$)', ' ', text)  
    # sequences of white spaces  
    text = re.sub(r'\s+', ' ', text)  
    return text.strip()