UTD Summer Workshop on NLP with Python

Assignment 1

Upload your completed Python program file to the MS Teams “Assignments” folder. Be sure to include your name in the comments at the top of the file in the following format:

```
################################################################################
#
#         FILE:
#           filename.py
#       AUTHOR:
#           Your Name
#  DESCRIPTION:
#           Description of your program, what it does
# DEPENDENCIES:
#           Created with Python 3.10.11 (Python version)
#           Any extra libraries required to run (like NLTK)
#
################################################################################
```

Submit a single file Python program that does the following:

Provided a plain text file (text_news.txt), your Python program should first split on paragraphs “<p>”, next on sentences, then extract dates referenced in the text as “date” objects. If no year is mentioned, assume the current calendar year. For each instance, your output should be to print a 2-tuple in the form,
    str
    [date list]
where the string str is the original sentence from which the date was extracted and [date list] is a list of dates in the sentence.

    Input: “Can we meet Sep 21st?”
    Output: "Can we meet Sep 21?"
        [2023-09-21]

Notes:

- Sentences should be delimited on periods, question marks, and exclamations. You can use the sentence tokenizer from NLTK
- Be sure to ignore periods used in abbreviations that are not sentence delimiters! e.g. Ms., Mr., `Dr., Prof., etc.

Your date parser should recognize the following date formats:

- Dates in the following five formats:
    - YYYY-MM-DD 2023-07-04
    - MM/DD/YYYY 07/04/2023
    - MM/DD/YY 07/04/23
    - BB DD, YYYY July 04, 2023
        - Also July 4, 2023
    - bb DD, YYYY Jul 04, 2023
        - Also Jul 4, 2023

You will need a regular expression to match each one of these.

For advanced students, you may also include the additional formats:

- Relative days: “today”, “yesterday”, “tomorrow”, “this Wednesday”, “next Thursday”, “last Friday”
    - “last” and “next” should reference the preceding and upcoming ISO weeks, respectively
    - “this *weekday*” should reference the 7 day span starting from today.