Jennifer Haliewicz
Ling 508 Project: Classes & tests
Ling 508: Prof. Eric Jackson
July 21, 2024

1. I have created a repository for the project at https://github.com/uazhlt-ms-program/

ling_508_project.git. It includes the class def file, test file, as well as the Dockerfile, Requirements, gitignore, etc.

2. The following pages include:
    a. The full code of classes_def.py
    b. The full code of test_def_classes.py
    c. A screenshot of the tests results from an html file output from pycharm.

```python
import nltk
from nltk.corpus import wordnet
from nltk.stem import WordNetLemmatizer
from nltk import pos_tag
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
from nltk.tokenize import sent_tokenize,
word_tokenize
import string

class Text:
    def __init__(self, text):
        self.text = text

    def split_into_sent(self):
        sentences = sent_tokenize(self.text)
        return sentences

    def __str__(self):
        return self.text

class Sent:
    def __init__(self, sentences):
        self.sentences = [sentence for sentence in
sentences]

    def get_sentence(self, index):
        if index < 0 or index >= len(self.sentences):
            raise IndexError("Index out of range")
        return self.sentences[index]

    def __str__(self):
        return ' '.join(self.sentences)

class LexEntry:
    wnl = WordNetLemmatizer()

    pos = {"noun", "pronoun", "verb", "adjective", "
adverb", "preposition", "determiner", "conjunction",
           "interjection", }

```

```
39      def __init__(self, sentence):
40          self.wnl = WordNetLemmatizer()
41          self.lex_entries = self.process_text(sentence
    ).split()
42
43      def process_text(self, text):
44          translator = str.maketrans('', '', string.
    punctuation)
45          cleaned_text = text.translate(translator)
46          return cleaned_text.lower()
47
48      def split_into_words(self):
49          return self.lex_entries
50
51      def get_word_len(self):
52          return [len(word) for word in self.
    lex_entries]
53
54      def get_word_index(self, word):
55          try:
56              return self.lex_entries.index(word)
57          except ValueError:
58              return -1
59
60      def get_pos(self, word):
61          tag = pos_tag([word])[0][1]
62          if tag.startswith('J'):
63              return wordnet.ADJ
64          elif tag.startswith('V'):
65              return wordnet.VERB
66          elif tag.startswith('N'):
67              return wordnet.NOUN
68          elif tag.startswith('R'):
69              return wordnet.ADV
70          else:
71              return wordnet.NOUN  # Default to noun if
     not found
72
73      def lemmatize_word(self, word):
74          pos = self.get_pos(word)
75          return self.wnl.lemmatize(word, pos)
```

```
76
77 text = Text("Hello world. This is a test sentence.")
78 sentences = text.split_into_sent()
79 print(sentences)
80
81 sent = Sent(sentences)
82 print(sent.get_sentence(1))
83
84 lex_entry = LexEntry(sent.get_sentence(1))
85 print(lex_entry.split_into_words())
86 print(lex_entry.get_word_len())
87 print(lex_entry.get_word_index("test"))
88 print(lex_entry.lemmatize_word("favoring"))
```

```python
import pytest
from classes_def import Text, Sent, LexEntry

def test_sentences():
    text = Text("Hello, this is an example text. It has multiple sentences.")
    sentences = text.split_into_sent()
    sent = Sent(sentences)

    print("Sentences:")
    for i in range(len(sentences)):
        try:
            print(f"Sentence {i}: {sent.get_sentence(i)}")
        except IndexError as e:
            print(e)

    assert sentences == ["Hello, this is an example text.",
                         "It has multiple sentences."], f"Expected sentences did not match: {sentences}"


def test_lex_entries():
    text = Text("Hello, this is an example text. It has multiple sentences.")
    sentences = text.split_into_sent()
    sent_chunk = Sent(sentences)
    for sentence in sentences:
        lex_entry = LexEntry(sentence)
        words = lex_entry.split_into_words()
        print(f"\nWords in '{sentence}':")
        for word in words:
            print(f"Word: {word}, Length: {len(word)}, Index: {lex_entry.get_word_index(word)}")

    first_sentence_lex = LexEntry(sentences[0])
    assert first_sentence_lex.split_into_words() == ["hello", "this", "is", "an", "example",

"text"], f"Expected words did not match: {
```

```
first_sentence_lex.split_into_words()}"

    second_sentence_lex = LexEntry(sentences[1])
    assert second_sentence_lex.split_into_words() == ["it", "has", "multiple",
 "sentences"], f"Expected words did not match: {second_sentence_lex.split_into_words()}"

def test_lemma():
    text = Text("Hello, this is an example text. It has multiple sentences.")
    sentences = text.split_into_sent()
    sent = Sent(sentences)

    for sentence in sentences:
        lex_entry = LexEntry(sentence)
        words = lex_entry.split_into_words()

        for word in words:
            pos = lex_entry.get_pos(word)
            lemma = lex_entry.lemmatize_word(word)

            # Print or assert as needed for testing
            print(f"Word: {word}, POS: {pos}, Lemma: {lemma}")

            # Example assertion (adjust as needed based on your expected outcomes)
            # This example assumes that 'example' should be lemmatized to 'example'
            if word == 'example':
                assert lemma == 'example', f"Expected 'example', but got '{lemma}'"
            elif word == 'has':
                assert lemma == 'have', f"Expected 'have', but got '{lemma}'"

if __name__ == "__main__":
    pytest.main()
```

pytest in test_def_classes.py: 3 total, 3 passed

10 ms

Collapse | Expand

test_def_classes

10 ms

test_sentences

passed 0 ms

PASSED [33%]Sentences:
Sentence 0: Hello, this is an example text.
Sentence 1: It has multiple sentences.

test_lex_entries

passed 0 ms

PASSED [66%]
Words in 'Hello, this is an example text.':
Word: hello, Length: 5, Index: 0
Word: this, Length: 4, Index: 1
Word: is, Length: 2, Index: 2
Word: an, Length: 2, Index: 3
Word: example, Length: 7, Index: 4
Word: text, Length: 4, Index: 5
Words in 'It has multiple sentences.':
Word: it, Length: 2, Index: 0
Word: has, Length: 3, Index: 1
Word: multiple, Length: 8, Index: 2
Word: sentences, Length: 9, Index: 3

test_lemma

passed 10 ms

PASSED [100%]Word: hello, POS: n, Lemma: hello
Word: this, POS: n, Lemma: this
Word: is, POS: v, Lemma: be
Word: an, POS: n, Lemma: an
Word: example, POS: n, Lemma: example
Word: text, POS: n, Lemma: text
Word: it, POS: n, Lemma: it
Word: has, POS: v, Lemma: have
Word: multiple, POS: n, Lemma: multiple
Word: sentences, POS: n, Lemma: sentence