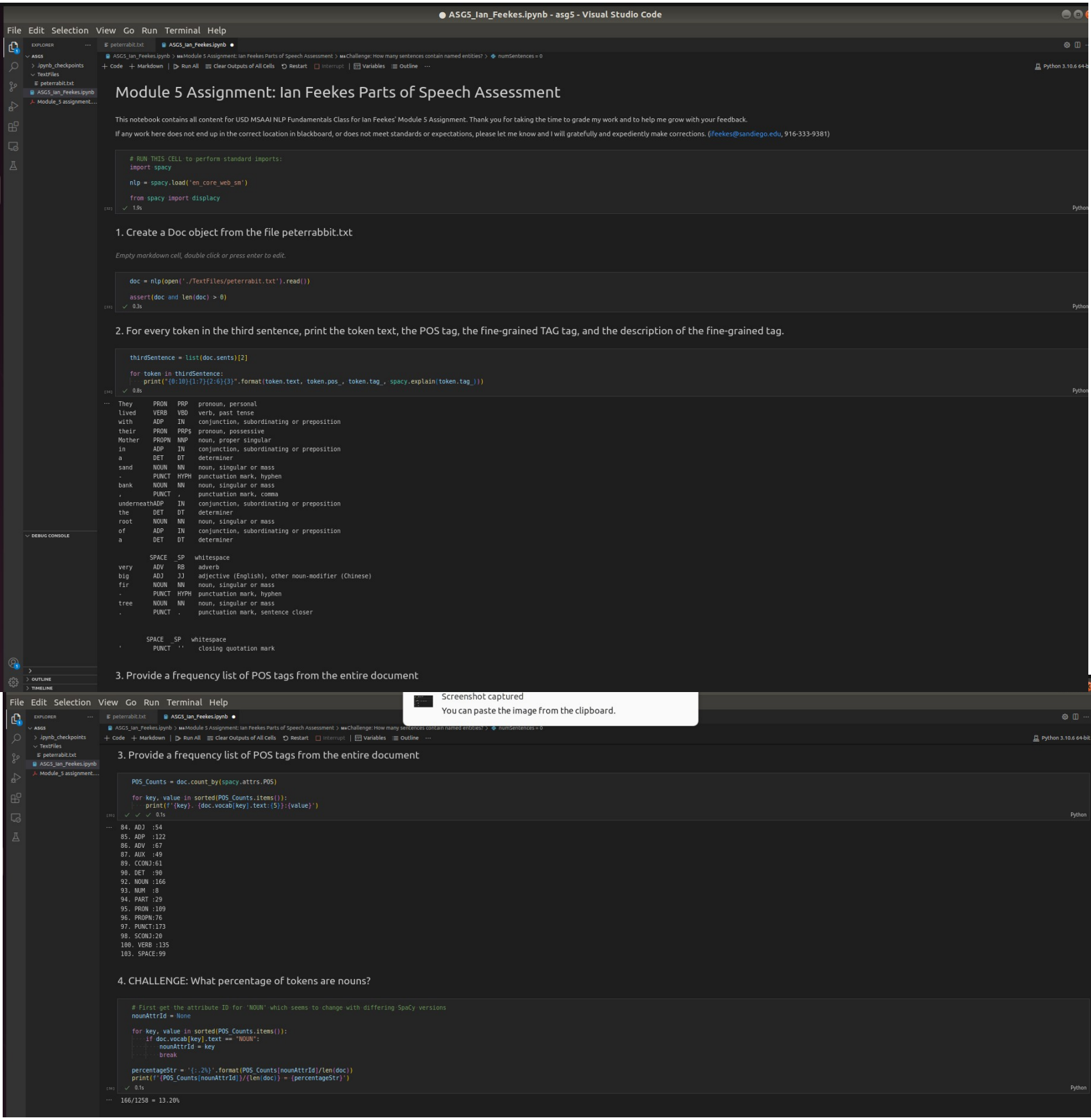

ASG5_Ian_Feekes.ipynb - asg5 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
peterrabbit.txt
ASG5_Ian_Feekes.ipynb
Code Markdown Run All Clear Outputs of All Cells Restart Interrupt Variables Outline
Python 3.10.6 64-bit
Module 5 Assignment: Ian Feekes Parts of Speech Assessment
This notebook contains all content for USD MSAAI NLP Fundamentals Class for Ian Feekes' Module 5 Assignment. Thank you for taking the time to grade my work and to help me grow with your feedback.
If any work here does not end up in the correct location in blackboard, or does not meet standards or expectations, please let me know and I will gratefully and expediently make corrections. (ifeekes@sandiego.edu, 916-333-9381)
# RUN THIS CELL to perform standard imports:
import spacy
nlp = spacy.load('en_core_web_sm')
from spacy import displacy
1.9s
Python
1. Create a Doc object from the file peterrabbit.txt
Empty markdown cell, double click or press enter to edit.
doc = nlp(open('./TextFiles/peterrabit.txt').read())
assert(doc and len(doc) > 0)
0.3s
Python
2. For every token in the third sentence, print the token text, the POS tag, the fine-grained TAG tag, and the description of the fine-grained tag.
thirdSentence = list(doc.sents)[2]
for token in thirdSentence:
print("{0:10}{1:7}{2:6}{3}".format(token.text, token.pos_, token.tag_, spacy.explain(token.tag_)))
0.8s
Python
They PRON PRP pronoun, personal
lived VERB VBD verb, past tense
with ADP IN conjunction, subordinating or preposition
their PRON PRP$ pronoun, possessive
Mother PROPN NNP noun, proper singular
in ADP IN conjunction, subordinating or preposition
a DET DT determiner
sand NOUN NN noun, singular or mass
- PUNCT HYPH punctuation mark, hyphen
bank NOUN NN noun, singular or mass
, PUNCT , punctuation mark, comma
underneathADP IN conjunction, subordinating or preposition
the DET DT determiner
root NOUN NN noun, singular or mass
of ADP IN conjunction, subordinating or preposition
a DET DT determiner
SPACE _SP whitespace
very ADV RB adverb
big ADJ JJ adjective (English), other noun-modifier (Chinese)
fir NOUN NN noun, singular or mass
- PUNCT HYPH punctuation mark, hyphen
tree NOUN NN noun, singular or mass
. PUNCT . punctuation mark, sentence closer
SPACE _SP whitespace
' PUNCT '' closing quotation mark
3. Provide a frequency list of POS tags from the entire document
Screenshot captured
You can paste the image from the clipboard.
3. Provide a frequency list of POS tags from the entire document
POS_Counts = doc.count_by(spacy.attrs.POS)
for key, value in sorted(POS_Counts.items()):
print(f'{key}. {doc.vocab[key].text:{5}}:{value}')
0.1s
Python
84. ADJ :54
85. ADP :122
86. ADV :67
87. AUX :49
89. CCONJ:61
90. DET :90
92. NOUN :166
93. NUM :8
94. PART :29
95. PRON :109
96. PROPN:76
97. PUNCT:173
98. SCONJ:20
100. VERB :135
103. SPACE:99
4. CHALLENGE: What percentage of tokens are nouns?
# First get the attribute ID for 'NOUN' which seems to change with differing SpaCy versions
nounAttrId = None
for key, value in sorted(POS_Counts.items()):
if doc.vocab[key].text == "NOUN":
nounAttrId = key
break
percentageStr = '{:.2%}'.format(POS_Counts[nounAttrId]/len(doc))
print(f'{POS_Counts[nounAttrId]}/{len(doc)} = {percentageStr}')
0.1s
Python
166/1258 = 13.20%

● ASG5_Ian_Feekes.ipynb - asg5 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

peterrabit.txt | ASG5_Ian_Feekes.ipynb ●

ASG5_Ian_Feekes.ipynb > Module 5 Assignment: Ian Feekes Parts of Speech Assessment > Challenge: How many sentences contain named entities? > numSentences = 0

+ Code + Markdown | Run All | Clear Outputs of All Cells | Restart | Interrupt | Variables | Outline | Python 3.10.6 64-bit

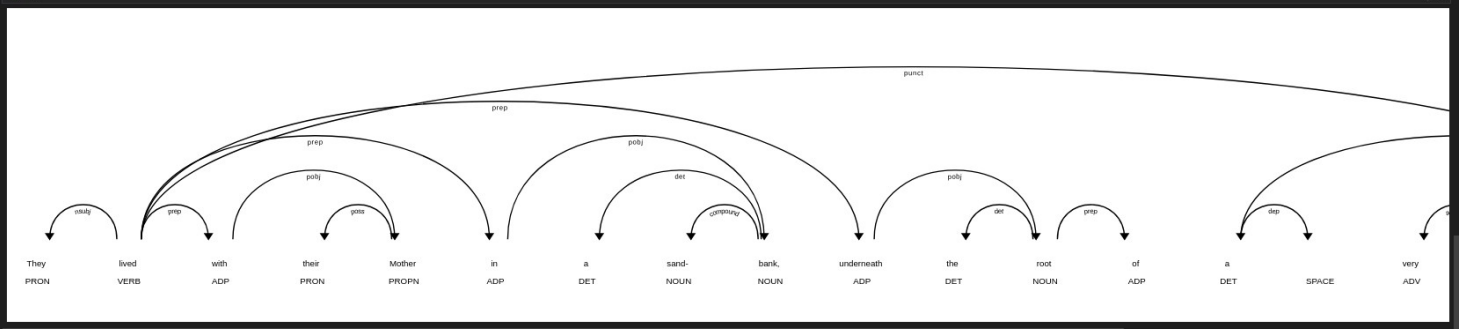
5. Display the Dependency Parse for the third sentence

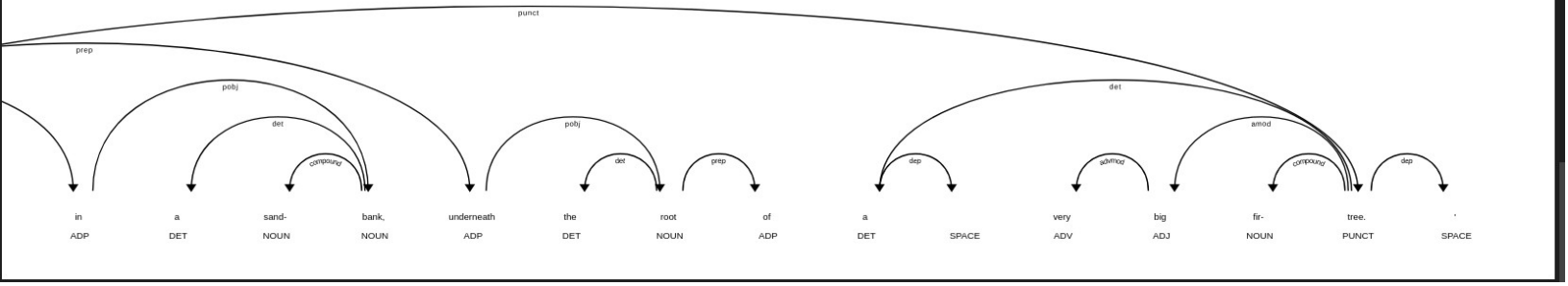
```
# import the displaCy library
from spacy import displacy

thirdSentenceDoc = nlp(thirdSentence.text)

displacy.render(thirdSentenceDoc, style="dep", jupyter= True, options={'distance': 150})
```

[37] ✓ 0.1s Python





● ASG5_Ian_Feekes.ipynb - asg5 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

peterrabit.txt | ASG5_Ian_Feekes.ipynb ●

ASG5_Ian_Feekes.ipynb > Module 5 Assignment: Ian Feekes Parts of Speech Assessment > Display the Dependency Parse for the third sentence > # import the displaCy library

+ Code + Markdown | Run All | Clear Outputs of All Cells | Restart | Interrupt | Variables | Outline | Python 3.10.6 64-bit

6. Show the first two named entities from Beatrix Potter's *The Tale of Peter Rabit*

```
def showEnts(doc, numEnts):
    if doc == None or doc.ents == None:
        return
    if numEnts < 0 or numEnts == None:
        numEnts = len(doc.ents)
    for ent in doc.ents[:numEnts]:
        print(ent.text + ' - ' + ent.label_ + ' - ' + str(spacy.explain(ent.label_)))

showEnts(doc, 2)
```

[38] ✓ 0.1s Python

```
The Tale of Peter Rabbit - WORK_OF_ART - Titles of books, songs, etc.
Beatrix Potter - PERSON - People, including fictional
```

7. How many sentences are contained in The Tale of Peter Rabbit?

```
len(list(doc.sents))
```

[39] ✓ 0.1s Python

```
55
```

8. Challenge: How many sentences contain named entities?

```
numSentences = 0
sentences = list(doc.sents)
sentenceIndex = 0

for sentence in sentences:
    for ent in doc.ents:
        if ent.text in sentence.text:
            numSentences = numSentences + 1
            break

numSentences
```

[40] ✓ 0.1s Python

```
36
```

9. CHALLENGE: Display the named entity visualization for list_of_sents[0] from the previous problem

+ Code + Markdown

```
displacy.render(sentences[0], style="ent", jupyter=True)
```

[41] ✓ 0.1s Python

The Tale of Peter Rabbit WORK_OF_ART , by Beatrix Potter PERSON (1902 DATE).