Name: Muhammad Ghazanfar

Roll#: 20P-0567

Sec: 7A

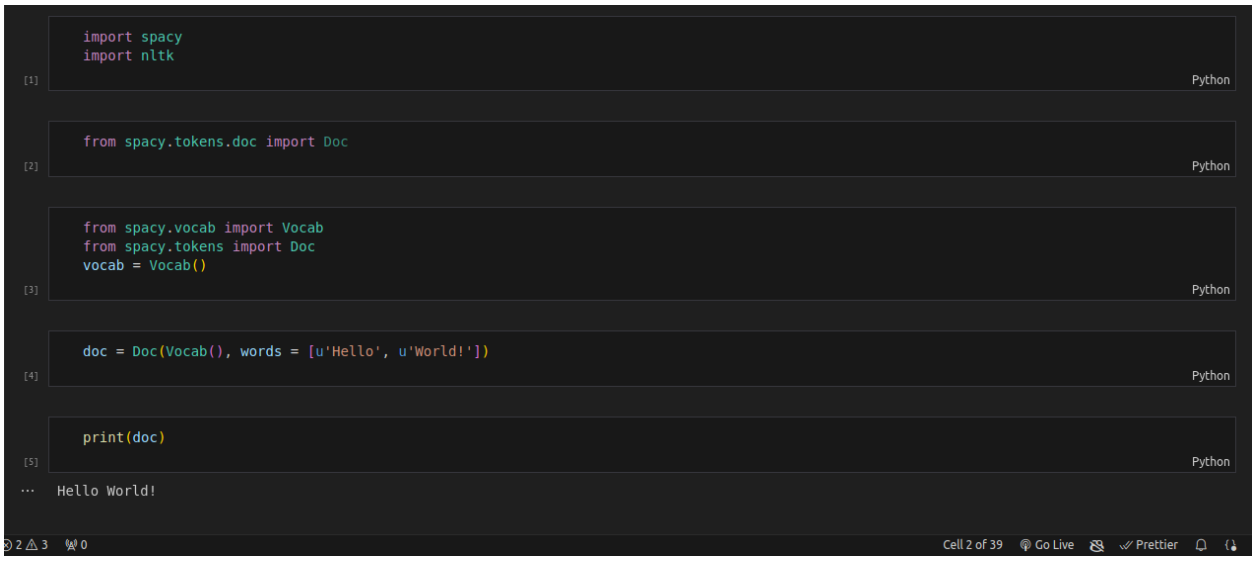
Course: NLP

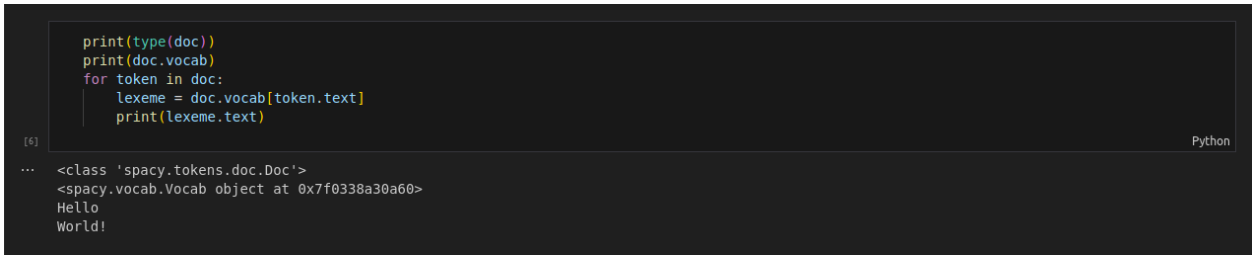
Lab#03

Task 1: Spacy Installation

Installations done, virtual environment created.

Task 2: spaCy Hello World





Attempt the following Questions:

1. The Vocab() object belongs to which class?

The `Vocab()` object belongs to the `spacy.vocab.Vocab` class in spaCy.

2. What is a Lexeme object (You need to check the API)

A Lexeme object is associated with a vocabulary entry that signifies a distinct word type. It encapsulates information about the word, such as its written form, lexical attributes, and linguistic features. These Lexeme objects are pivotal for storing and accessing details about specific words, and they are created as part of the tokenization procedure. To enhance memory efficiency, Lexeme objects are utilized universally for all instances of the same word throughout a text.

```python
nlp = spacy.load("en_core_web_sm")
doc = nlp(u'I want to learn spaCy.')
token_text1 = [token.text for token in doc]
token_text2 = [doc[i].text for i in range(len(doc))]
print(token_text1)
print(token_text2)
```

[10] Python

```
['I', 'want', 'to', 'learn', 'spaCy', '.']
['I', 'want', 'to', 'learn', 'spaCy', '.']
```

Attempt the following Questions:

1. What is en_core_web_sm?

en_core_web_sm refers to a pre-trained language model designed for the English language within spaCy, a widely-used open-source library for Natural Language Processing (NLP). It serves as a compact and efficient model optimized for CPU usage, making it particularly suitable for tasks where speed and resource constraints are critical.

Components:

Tokenizer: Segments text into individual words and punctuation marks.
Tagger: Assigns parts-of-speech (POS) tags to each token (e.g., noun, verb, adjective).
Parser: Analyzes the grammatical structure of sentences.
Named Entity Recognition (NER): Identifies and categorizes named entities such as people, places, and organizations.
Attribute Ruler: Applies custom rules to extract additional information from the text.
Lemmatizer: Reduces words to their base form (e.g., "running" to "run").

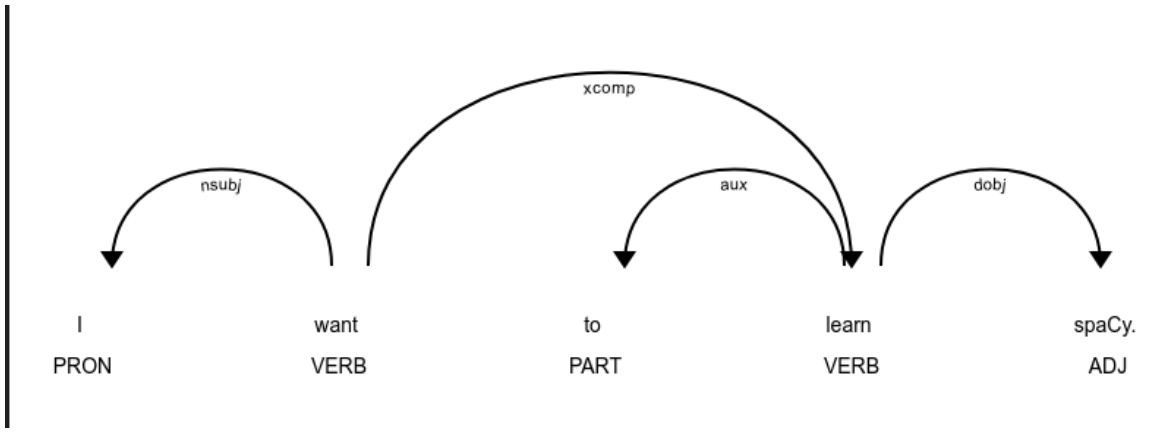
2. What is the size of en_core_web_sm?

The size of en_core_web_sm is approximately 12 MB, making it a relatively lightweight model compared to other language models. This modest size facilitates quicker downloading and deployment, especially on smaller systems.

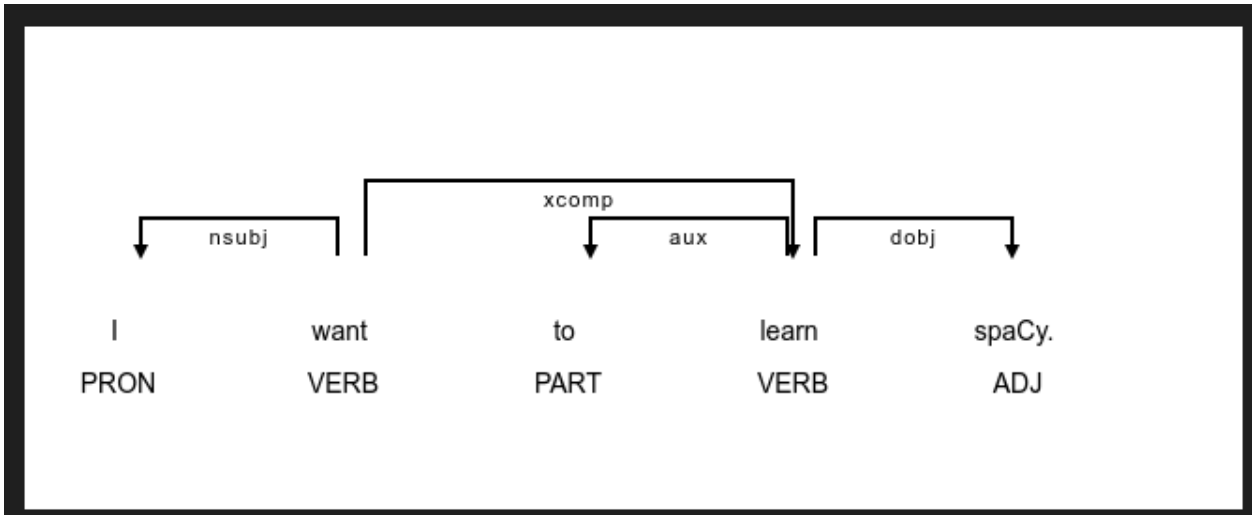
3. What other variations can be used?

spaCy offers various versions of English language models with different sizes and capabilities. Some notable variations include:
en_core_web_md (Medium): A medium-sized English model with more vectors, potentially providing increased accuracy. However, it comes with a larger file size compared to en_core_web_sm.
en_core_web_lg (Large): A large English model featuring even more vectors, offering higher accuracy. It stands as the largest among the web models in terms of size.
Custom models: In addition to pre-trained models, spaCy empowers users to train custom models tailored to their specific domains or tasks using their own datasets.

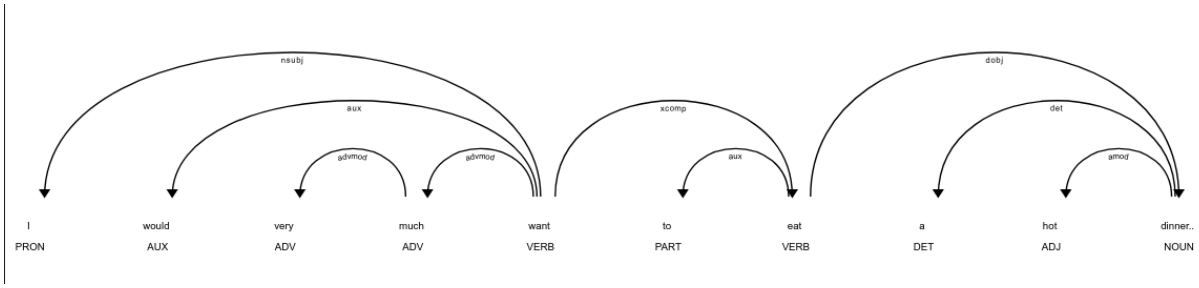
1. **Draw the left and right dependencies for the sentence: I want to learn spaCy.**



2. **Draw the children for the sentence: I want to learn spaCy.**



3. **Draw the left and right dependencies for the sentence: I would very much want to eat a hot dinner.**



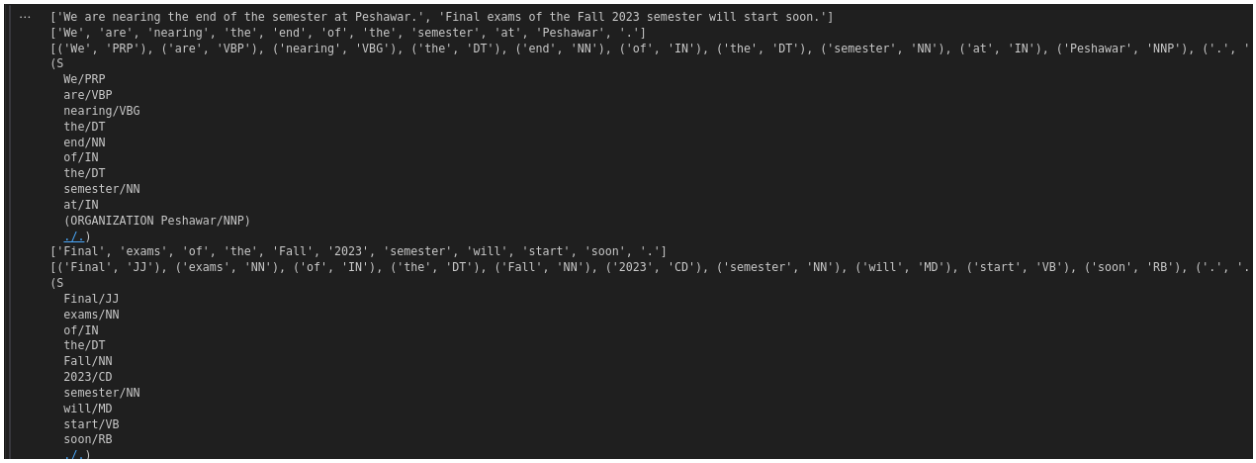
4. Present a list of all dependency grammars of your sentences above.

```
Dependency Grammars:
Sentence#1
I --(nsubj)--> want
want --(ROOT)--> want
to --(aux)--> learn
learn --(xcomp)--> want
spaCy --(dobj)--> learn
. --(punct)--> want

========================================


Dependency Grammars:
Sentence#2
I --(nsubj)--> want
want --(ROOT)--> want
to --(aux)--> learn
learn --(xcomp)--> want
spaCy --(dobj)--> learn
. --(punct)--> want

========================================


Dependency Grammars:
Sentence#3
I --(nsubj)--> want
would --(aux)--> want
very --(advmod)--> much
much --(advmod)--> want
want --(ROOT)--> want
to --(aux)--> eat
eat --(xcomp)--> want
a --(det)--> dinner
hot --(amod)--> dinner
dinner --(dobj)--> eat
. --(punct)--> want
```

Task 3: NLTK vs spaCy Pipelines

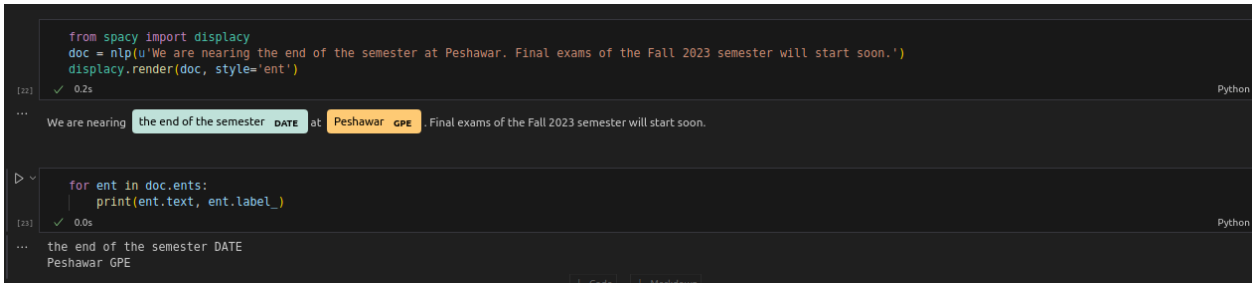
Attempt the following Questions:

What did the Named Entity Output of the NLTK pipeline look like? Present its output.



```
['We are nearing the end of the semester at Peshawar.', 'Final exams of the Fall 2023 semester will start soon.']
['We', 'are', 'nearing', 'the', 'end', 'of', 'the', 'semester', 'at', 'Peshawar', '.']
[('We', 'PRP'), ('are', 'VBP'), ('nearing', 'VBG'), ('the', 'DT'), ('end', 'NN'), ('of', 'IN'), ('the', 'DT'), ('semester', 'NN'), ('at', 'IN'), ('Peshawar', 'NNP'), ('.', '
(S
  We/PRP
  are/VBP
  nearing/VBG
  the/DT
  end/NN
  of/IN
  the/DT
  semester/NN
  at/IN
  (ORGANIZATION Peshawar/NNP)
  ./.)
['Final', 'exams', 'of', 'the', 'Fall', '2023', 'semester', 'will', 'start', 'soon', '.']
[('Final', 'JJ'), ('exams', 'NN'), ('of', 'IN'), ('the', 'DT'), ('Fall', 'NN'), ('2023', 'CD'), ('semester', 'NN'), ('will', 'MD'), ('start', 'VB'), ('soon', 'RB'), ('.', '.
(S
  Final/JJ
  exams/NN
  of/IN
  the/DT
  Fall/NN
  2023/CD
  semester/NN
  will/MD
  start/VB
  soon/RB
  ./.)
```

What did the Named Entity Output of the spaCy pipeline look like? Present its output.



```
from spacy import displacy
doc = nlp(u'We are nearing the end of the semester at Peshawar. Final exams of the Fall 2023 semester will start soon.')
displacy.render(doc, style='ent')
```

[22] ✓ 0.2s Python

We are nearing the end of the semester DATE at Peshawar GPE . Final exams of the Fall 2023 semester will start soon.

```
for ent in doc.ents:
    print(ent.text, ent.label_)
```

[23] ✓ 0.0s Python

```
the end of the semester DATE
Peshawar GPE
```

Attempt the following Questions:

1. What is the default pipeline structure of spaCy?

```
# Check the pipeline components before any modifications
print("Pipeline components before:", nlp.pipe_names)
```

[26] ✓ 0.0s Python

```
Pipeline components before: ['tok2vec', 'tagger', 'attribute_ruler', 'lemmatizer', 'ner']
```

Task 4: Finding Patterns in Sentences

```
doc = nlp(u'I want to learn spaCy.')
displacy.render(doc, style='dep')
```

[32] ✓ 0.0s Python

```
/home/ghazanfar/.local/lib/python3.10/site-packages/spacy/displacy/__init__.py:141: UserWarning: [W005] Doc object not parsed. This means displaCy won't be able to generate a
https://spacy.io/usage/models
  warnings.warn(Warnings.W005)
```

I	want	to	learn	spaCy.
PRON	VERB	PART	VERB	ADJ

```
doc = nlp(u'How do I learn spaCy.')
displacy.render(doc, style='dep')
```

[33] ✓ 0.0s Python

How	do	I	learn	spaCy.
SCONJ	AUX	PRON	VERB	ADJ

```
import spacy
nlp = spacy.load('en_core_web_sm')
def dep_pattern(doc):
    for i in range(len(doc)-1):
        print(doc[i].dep_)

        if doc[i].dep_ == 'nsubj' and doc[i+1].dep_ == 'ROOT' and doc[i+2].dep_ =='acomp':
            return True
    return False
doc = nlp(u'How do I learn spaCy.')
if dep_pattern(doc):

    print('Found')

else:

    print('Not found')
```

[34] ✓ 1.0s Python

```
advmod
aux
nsubj
Found
```