Week 9: Dependency Parsing using Stanza

Step 1: Install Stanza

```
pip install stanza
```

Step 2: Download English model

```
import stanza
stanza.download('en')  # Run this only once
```

9.1. Simple Dependency Parsing Program

```
import stanza

# Load English pipeline
nlp = stanza.Pipeline('en')

# Input sentence
sentence = "The quick brown fox jumps over the lazy dog."

# Process the sentence
doc = nlp(sentence)

# Print dependencies
print("Word\tHead\tRelation")
for sent in doc.sentences:
    for word in sent.words:
        head = sent.words[word.head - 1].text if word.head > 0 else "ROOT"
        print(f"{word.text}\t{head}\t{word.deprel}")
```

Output Example:

```
Word    Head    Relation
The     fox     det
quick   fox     amod
brown   fox     amod
fox     jumps   nsubj
jumps   ROOT    root
over    jumps   case
the     dog     det
lazy    dog     amod
dog     over    obl
.       jumps   punct
```

9.2. Advanced Dependency Tree Visualization using `networkx` + `matplotlib`

```
import stanza
import networkx as nx
import matplotlib.pyplot as plt
```

```
# Load NLP pipeline
nlp = stanza.Pipeline('en')

sentence = "The quick brown fox jumps over the lazy dog."
doc = nlp(sentence)

G = nx.DiGraph()

for sent in doc.sentences:
    for word in sent.words:
        head_text = "ROOT" if word.head == 0 else sent.words[word.head -
1].text
        G.add_edge(head_text, word.text, label=word.deprel)

# Draw the graph
pos = nx.spring_layout(G)
labels = nx.get_edge_attributes(G, 'label')
nx.draw(G, pos, with_labels=True, node_size=2000, node_color='lightblue',
font_size=10)
nx.draw_networkx_edge_labels(G, pos, edge_labels=labels, font_color='red')
plt.title("Dependency Parse Tree")
plt.show()
```