Name: Sanket S. Fulzele

Roll No: 371018

PRN: 22110728

Div: A

Comparison between Deep parsing and Shallow Parsing

Here's a brief discussion comparing deep parsing and shallow parsing in tabular form:

Aspect	Deep Parsing	Shallow Parsing
Scope	Analyzes the entire sentence or discourse	Focuses on smaller syntactic units or phrases
Granularity	Fine-grained analysis of sentence structure	Coarser analysis, often based on parts of speech tags or chunks
Information	Captures detailed syntactic relationships	Emphasizes on identifying phrases and entities
Complexity	More computationally intensive	Less computationally intensive
Output	Produces parse trees or dependency graphs	Outputs chunks or phrases
Applications	Machine translation, syntax-driven applications	Information extraction, named entity recognition

Implementation

```
import nltk
# nltk.download('maxent_ne_chunker')
# nltk.download('words')

# Customized Grammar
custom_grammar = r"""
  NP: {<DT>?<JJ>*<NN>}  # Noun Phrase
  AP: {<RB>?<JJ>+}      # Adjective Phrase
  AdvP: {<RB>+}         # Adverb Phrase
  VP: {<VB.*><NP|PP>*}  # Verb Phrase
"""
# Create a chunk parser with the customized grammar
chunk_parser = nltk.RegexpParser(custom_grammar)

# Sample sentence
sentence = [("The", "DT"), ("quick", "JJ"), ("brown", "JJ"), ("fox", "NN"), ("jumps", "VBZ"), ("over", "IN"),
            ("the", "DT"), ("lazy", "JJ"), ("dog", "NN")]

# Perform chunking using the custom grammar
chunks_custom = chunk_parser.parse(sentence)
print("Customized Chunking:", chunks_custom)

# Perform chunking using inbuilt functions

chunks_builtin = nltk.ne_chunk(sentence)
print("Built-in Chunking:", chunks_builtin)

# Compare outputs
print("\nCustomized Chunking:")
for subtree in chunks_custom.subtrees():
    if subtree.label() in ['NP', 'AP', 'AdvP', 'VP']:
        print(subtree.label(), ":", " ".join(word for word, pos in subtree.leaves()))

print("\nBuilt-in Chunking:")
print(chunks_builtin)
```

```
Customized Chunking: (S
  (NP The/DT quick/JJ brown/JJ fox/NN)
  (VP jumps/VBZ)
  over/IN
  (NP the/DT lazy/JJ dog/NN))
Built-in Chunking: (S
  The/DT
  quick/JJ
  brown/JJ
  fox/NN
  jumps/VBZ
  over/IN
  the/DT
  lazy/JJ
  dog/NN)

Customized Chunking:
NP : The quick brown fox
VP : jumps
NP : the lazy dog

Built-in Chunking:
(S
  The/DT
  quick/JJ
  brown/JJ
  fox/NN
  jumps/VBZ
  over/IN
  the/DT
  lazy/JJ
  dog/NN)
```

By comparing the outputs, you can analyze how well your customized grammar captures the desired chunks compared to NLTK's built-in chunking functions. Adjustments to the grammar and patterns may be needed based on the specific linguistic structures you want to capture.

```
Start coding or generate with AI.
```