

AI Lab\Exp_09\partsOfspeech.py

```
1 import nltk
2 from nltk import pos_tag
3 from nltk.tokenize import word_tokenize
4
5 def ensure_nltk_resources():
6     resources = [
7         ('tokenizers/punkt', 'punkt'),
8         ('tokenizers/punkt_tab', 'punkt_tab'),
9         ('taggers/averaged_perceptron_tagger', 'averaged_perceptron_tagger'),
10        ('taggers/averaged_perceptron_tagger_eng', 'averaged_perceptron-
11        tagger_eng')
12    ]
13
14    print("Checking NLTK resources...")
15    for path, resource in resources:
16        try:
17            nltk.data.find(path)
18        except LookupError:
19            print(f"Downloading missing resource: {resource}")
20            try:
21                nltk.download(resource, quiet=True)
22            except Exception as e:
23                print(f"Note: Could not download {resource}. Error: {e}")
24
25 def get_pos_tags(sentence):
26     tokens = word_tokenize(sentence)
27     tagged_tokens = pos_tag(tokens)
28     return tagged_tokens
29
30 def print_explained_tags(tagged_tokens):
31     tag_descriptions = {
32         'CC': 'Coordinating conjunction',
33         'CD': 'Cardinal number',
34         'DT': 'Determiner',
35         'EX': 'Existential there',
36         'IN': 'Preposition or subordinating conjunction',
37         'JJ': 'Adjective',
38         'JJR': 'Adjective, comparative',
39         'JJS': 'Adjective, superlative',
40         'NN': 'Noun, singular or mass',
41         'NNS': 'Noun, plural',
42         'NNP': 'Proper noun, singular',
43         'NNPS': 'Proper noun, plural',
44         'RB': 'Adverb',
45         'RBR': 'Adverb, comparative',
46         'RBS': 'Adverb, superlative',
47         'VB': 'Verb, base form',
48         'VBD': 'Verb, past tense',
49         'VBG': 'Verb, gerund or present participle',
50         'VBN': 'Verb, past participle',
51         'VBP': 'Verb, non-3rd person singular present',
52         'VBZ': 'Verb, 3rd person singular present',
```

```
52     }
53
54     print(f"\n{'WORD':<15} {'TAG':<10} {'DESCRIPTION'}")
55     print("-" * 50)
56
57     for word, tag in tagged_tokens:
58         description = tag_descriptions.get(tag, "Other/Special Symbol")
59         print(f"{word:<15} {tag:<10} {description}")
60
61 if __name__ == "__main__":
62     ensure_nltk_resources()
63
64     text = "The quick brown fox jumps over the lazy dog."
65
66     print(f"\nProcessing sentence: \"{text}\"")
67
68     tags = get_pos_tags(text)
69
70     print("\nRaw Output (List of Tuples):")
71     print(tags)
72
73     print_explained_tags(tags)
```