

## AI Lab\Exp\_09\parts0fspeech.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_
tagger_eng')
11    ]
12
13    print("Checking NLTK resources...")
14    for path, resource in resources:
15        try:
16            nltk.data.find(path)
17        except LookupError:
18            print(f"Downloading missing resource: {resource}")
19            try:
20                nltk.download(resource, quiet=True)
21            except Exception as e:
22                print(f"Note: Could not download {resource}. Error: {e}")
23
24 def get_pos_tags(sentence):
25     tokens = word_tokenize(sentence)
26     tagged_tokens = pos_tag(tokens)
27     return tagged_tokens
28
29 def print_explained_tags(tagged_tokens):
30     tag_descriptions = {
31         'CC': 'Coordinating conjunction',
32         'CD': 'Cardinal number',
33         'DT': 'Determiner',
34         'EX': 'Existential there',
35         'IN': 'Preposition or subordinating conjunction',
36         'JJ': 'Adjective',
37         'JJR': 'Adjective, comparative',
38         'JJS': 'Adjective, superlative',
39         'NN': 'Noun, singular or mass',
40         'NNS': 'Noun, plural',
41         'NNP': 'Proper noun, singular',
42         'NNPS': 'Proper noun, plural',
43         'RB': 'Adverb',
44         'RBR': 'Adverb, comparative',
45         'RBS': 'Adverb, superlative',
46         'VB': 'Verb, base form',
47         'VBD': 'Verb, past tense',
48         'VBG': 'Verb, gerund or present participle',
49         'VBN': 'Verb, past participle',
50         'VBP': 'Verb, non-3rd person singular present',
51         '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)
```