Bansilal Ramnath Agarwal Charitable Trust's
Vishwakarma Institute of Information Technology

Department of
Artificial Intelligence and Data Science

Name: **Siddhesh Dilip Khairnar**		
Class: **TY**	Division: **B**	Roll No: **372028**
Semester: **6th**	Academic Year: **2023-24**	
Subject Name & Code: **Natural Language and Processing & ADUA32203**		
Title of Assignment: **Perform PoS tagging using regular expressions and inbuilt PoS taggers**		
Date of Performance: **09-02-2024**	Date of Submission: **16-03-2024**	

ASSIGNMENT NO: - 4

Aim: Perform PoS tagging using regular expressions and inbuilt PoS taggers

❖ THEORY:

1) Brief Discussion on PoS Tag Sets for Different Languages:

- Part-of-speech (PoS) tagging is a fundamental task in natural language processing (NLP) that involves labelling each word in a sentence with its corresponding part of speech, such as noun, verb, adjective, etc.
- PoS tag sets vary across different languages due to linguistic differences, morphological complexities, and syntactic structures.
- Each language presents its own set of challenges and nuances for PoS tagging, and researchers continue to develop and refine PoS tag sets and tagging algorithms to improve accuracy and efficiency across different languages.

2) Brief Discussion on Various Approaches for PoS Tagging:

- Part-of-speech (PoS) tagging is a crucial task in natural language processing (NLP) that involves assigning grammatical categories (such as noun, verb, adjective, etc.) to each word in a text.
- Various approaches have been developed over the years to tackle PoS tagging, each with its own advantages and limitations.
- Here's a brief discussion on some of these approaches:

1) Rule-Based Approach:

- Rule-based PoS tagging relies on hand-crafted linguistic rules to assign tags to words based on patterns in their context.
- These rules may consider features such as word suffixes, prefixes, and neighbouring words.
- While rule-based approaches can be transparent and interpretable,
  They often require extensive linguistic expertise and may not generalize well to diverse text genres or languages.

2) Probabilistic Approach:

- Probabilistic methods, such as Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs), model the likelihood of observing a sequence of tags given the input words.
- These models use training data to learn the probabilities of transitions between tags and the emission probabilities of words given tags.
- Probabilistic approaches are widely used for PoS tagging due to their flexibility, scalability, and ability to capture complex dependencies in language.

3) Hybrid Approaches:

- Hybrid approaches combine the strengths of rule-based, probabilistic, and deep learning methods to improve PoS tagging accuracy.
- For example, a hybrid system may use rule-based pre-processing to handle specific linguistic phenomena, followed by a probabilistic or deep learning model for tagging.
- These hybrid systems aim to leverage the advantages of different approaches while mitigating their individual weaknesses.

Each approach has its own set of trade-offs in terms of accuracy, efficiency, interpretability, and resource requirements. Researchers continue to explore novel techniques and combinations of methods to further advance PoS tagging performance across different languages and domains.

A) POS tagging using regular expression (English):

```python
import nltk
from nltk.tokenize import word_tokenize

# Define a dictionary to map POS tags to their descriptions
pos_tag_descriptions = {
    'CC': 'coordinating conjunction',
    'CD': 'cardinal digit',
    'DT': 'determiner',
    'EX': 'existential there',
    'FW': 'foreign word',
    'IN': 'preposition/subordinating conjunction',
    'JJ': 'adjective',
    'JJR': 'adjective, comparative',
    'JJS': 'adjective, superlative',
    'LS': 'list marker',
    'MD': 'modal',
    'NN': 'noun, singular or mass',
    'NNS': 'noun plural',
    'NNP': 'proper noun, singular',
    'NNPS': 'proper noun, plural',
    'PDT': 'predeterminer',
    'POS': 'possessive ending',
    'PRP': 'personal pronoun',
    'PRP$': 'possessive pronoun',
    'RB': 'adverb',
    'RBR': 'adverb, comparative',
    'RBS': 'adverb, superlative',
    'RP': 'particle',
    'SYM': 'symbol',
    'TO': 'to',
```

```
    'UH': 'interjection',
    'VB': 'verb, base form',
    'VBD': 'verb, past tense',
    'VBG': 'verb, gerund/present participle',
    'VBN': 'verb, past participle',
    'VBP': 'verb, sing. present, non-3d',
    'VBZ': 'verb, 3rd person sing. present',
    'WDT': 'wh-determiner',
    'WP': 'wh-pronoun',
    'WP$': 'possessive wh-pronoun',
    'WRB': 'wh-adverb'
}

def pos_tagging(sentence):
    # Tokenize the sentence into words
    words = word_tokenize(sentence)

    # Perform POS tagging
    pos_tags = nltk.pos_tag(words)

    # Map POS tags to descriptions
    tagged_words = [(word, pos, pos_tag_descriptions.get(pos,
    'Unknown')) for word, pos in pos_tags]

    return tagged_words

# Take input from the user
sentence = input("Enter a sentence: ")

# Perform POS tagging on the user input
tags = pos_tagging(sentence)

# Print tagged words with descriptions
print("Tagged Words with Descriptions:")
for word, pos, description in tags:
    print(f"Word: {word}, POS Tag: {pos}, Description:
    {description}")
```

Output:

```
Enter a sentence: The quick brown fox jumps over the lazy dog
Tagged Words with Descriptions:
Word: The, POS Tag: DT, Description: determiner
Word: quick, POS Tag: JJ, Description: adjective
Word: brown, POS Tag: NN, Description: noun, singular or mass
Word: fox, POS Tag: NN, Description: noun, singular or mass
Word: jumps, POS Tag: VBZ, Description: verb, 3rd person sing. present
Word: over, POS Tag: IN, Description: preposition/subordinating conjunction
Word: the, POS Tag: DT, Description: determiner
Word: lazy, POS Tag: JJ, Description: adjective
Word: dog, POS Tag: NN, Description: noun, singular or mass
```

B) POS tagging using inbuilt functions (English):

```python
import spacy

def pos_tagging_spacy(sentence):
    # Load the English language model in spaCy
    nlp = spacy.load("en_core_web_sm")

    # Process the sentence with the model
    doc = nlp(sentence)

    # Get the POS tags and their descriptions
    tagged_words = [(token.text, token.pos_, spacy.explain(token.
    pos_)) for token in doc]

    return tagged_words

# Take input from the user
sentence = input("Enter a sentence: ")

# Perform POS tagging on the user input
tags_spacy = pos_tagging_spacy(sentence)

# Print tagged words with descriptions
for word, pos, description in tags_spacy:
    print(f"Word: {word}, POS Tag: {pos}, Description:
    {description}")
```

Output:

```
Enter a sentence: The quick brown fox jumps over the lazy dog
Word: The, POS Tag: DET, Description: determiner
Word: quick, POS Tag: ADJ, Description: adjective
Word: brown, POS Tag: ADJ, Description: adjective
Word: fox, POS Tag: NOUN, Description: noun
Word: jumps, POS Tag: VERB, Description: verb
Word: over, POS Tag: ADP, Description: adposition
Word: the, POS Tag: DET, Description: determiner
Word: lazy, POS Tag: ADJ, Description: adjective
Word: dog, POS Tag: NOUN, Description: noun
```

C) POS tagging using regular expression (Marathi):

```python
import nltk
from nltk.tokenize import word_tokenize
from nltk.tag import tnt

# Marathi POS Tagset
marathi_pos_tagset = {
    'PRP': 'pronoun',
    'NN': 'noun',
    'QF': 'quantifier',
    'VRB': 'verb',
    'SYM': 'symbol',
    'CC': 'conjunction',
    'DET': 'determiner',
    'INTF': 'interjection',
    'NUM': 'numeral',
    'PSP': 'postposition',
    'RP': 'particle',
    'WQ': 'wh-word',
    'JJ': 'adjective',
    'FW': 'foreign word'
}

```

```python
# Train POS Tagger for Marathi
def train_marathi_tagger():
    marathi_data = [[('मला', 'PRP'), ('बाल', 'NN'), ('पशू', 'NN'), ('आवडतात', 'VRB')],
                    [('तुम्हाला', 'PRP'), ('कसं', 'QF'), ('वाटतं', 'VRB'), ('?', 'SYM')]]
    tagger = tnt.TnT()
    tagger.train(marathi_data)
    return tagger

# Tag Marathi Sentences
def pos_tagging_marathi(sentence, tagger):
    words = word_tokenize(sentence)
    tags = tagger.tag(words)
    return tags

# Sample Marathi Sentences
marathi_sentences = ["मला बाल पशूआवडतात", "तुमच्या कामाला जबाबदारी द्यावी मी नको", "तुम्हाला कसंवाटतं ?"]

# Train Marathi POS Tagger
marathi_tagger = train_marathi_tagger()

# Tag Marathi Sentences
for sent in marathi_sentences:
    print(pos_tagging_marathi(sent, marathi_tagger))
```

Output:

```
[('मला', 'PRP'), ('बाल', 'NN'), ('पशूआवडतात', 'Unk')]
[('तुमच्या', 'Unk'), ('कामाला', 'Unk'), ('जबाबदारी', 'Unk'), ('द्यावी', 'Unk'), ('मी', 'Unk'), ('नको', 'Unk')]
[('तुम्हाला', 'PRP'), ('कसंवाटतं', 'Unk'), ('?', 'SYM')]
```

Conclusion: Thus, we have successfully performed PoS tagging using inbuilt PoS taggers(for English and Marathi), regular expression, dictionary based PoS tagging and N-gram model based PoS tagging.