SATHYABAMA INSTITUTE OF SCIENCE & TECHNOLOGY

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SCSA 2604 NATURAL LANGUAGE PROCESSING LAB

LAB 6: CASE STUDY

AIM: Parts of Speech Tagging

Problem Statement:

An online news aggregator wants to improve its recommendation system by analyzing the content of news articles. To achieve this, they need to perform parts of speech tagging on the article text to extract relevant information such as key topics, sentiments, and entities mentioned.

Objectives :

1. Develop a parts of speech tagging system to analyze the content of news articles.
2. Extract key information such as nouns, verbs, adjectives, and other parts of speech to understand the structure of the articles.
3. Enhance the recommendation system by incorporating the extracted information to provide more accurate and personalized recommendations to users.

Dataset:

The dataset consists of a collection of news articles in text format. Each article is labeled with its category (e.g., politics, sports, entertainment) and contains textual content for analysis.

Approach:

1. Preprocess the dataset by tokenizing the text into words and sentences.
2. Perform parts of speech tagging using a pre-trained model or a custom-trained model.
3. Extract relevant parts of speech such as nouns, verbs, adjectives, and adverbs from the tagged text.
4. Analyze the distribution of different parts of speech across the articles to understand their linguistic characteristics.
5. Integrate the extracted information into the recommendation system to improve the relevance of recommended articles for users.

Program :

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

# Download NLTK resources (if not already downloaded)
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

def pos_tagging(text):
    sentences = sent_tokenize(text)
    tagged_tokens = []
    for sentence in sentences:
        tokens = word_tokenize(sentence)
        tagged_tokens.extend(nltk.pos_tag(tokens))
    return tagged_tokens

def main():
    # Example news article
    article_text = """
    Manchester United secured a 3-1 victory over Chelsea in yesterday's
match.
    Goals from Rashford, Greenwood, and Fernandes sealed the win for
United.
    Chelsea's only goal came from Pulisic in the first half.
    The victory boosts United's chances in the Premier League title
race.
    """

    tagged_tokens = pos_tagging(article_text)
    print("Original Article Text:\n", article_text)
    print("\nParts of Speech Tagging:")
    for token, pos_tag in tagged_tokens:
        print(f"{token}: {pos_tag}")

if __name__ == "__main__":
    main()
```

Output:

```
Original Article Text:

  Manchester United secured a 3-1 victory over Chelsea in yesterday's match.
  Goals from Rashford, Greenwood, and Fernandes sealed the win for United.
  Chelsea's only goal came from Pulisic in the first half.
  The victory boosts United's chances in the Premier League title race.


Parts of Speech Tagging:
```

```
Manchester: NNP
United: NNP
secured: VBD
a: DT
3-1: JJ
victory: NN
over: IN
Chelsea: NNP
in: IN
yesterday: NN
's: POS
match: NN
.: .
Goals: NNS
from: IN
Rashford: NNP
,: ,
Greenwood: NNP
,: ,
and: CC
Fernandes: NNP
sealed: VBD
the: DT
win: NN
for: IN
United: NNP
.: .
Chelsea: NN
's: POS
only: JJ
goal: NN
came: VBD
from: IN
Pulisic: NNP
in: IN
the: DT
first: JJ
half: NN
.: .
The: DT
victory: NN
boosts: VBZ
United: NNP
's: POS
chances: NNS
in: IN
the: DT
Premier: NNP
League: NNP
title: NN
race: NN
.: .
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

Result:

This program demonstrates the parts of speech tagging process on a sample news article. Each word in the article is followed by its corresponding part of speech tag. This information can be further utilized for analysis and decision-making in the recommendation system.