Mini Project

POS Taggers for Indian Languages

1. Introduction

Part-of-Speech (POS) tagging is a fundamental task in Natural Language Processing (NLP) that involves assigning grammatical categories—such as noun, verb, adjective, etc.—to each word in a sentence. In this mini project, we focus on implementing a POS tagger for Indian languages (specifically Hindi) using Python, Indic NLP Library for tokenization, and NLTK for tagging.

2. Objective

To develop a POS tagging system for Indian languages using basic NLP techniques. The goal is to tokenize input sentences and tag each word with its respective part-of-speech using a trained N-gram-based model.

3. Tools and Libraries Used

- Python 3

- NLTK (Natural Language Toolkit) – for tagging and corpus handling

- Indic NLP Library – for language-specific tokenization (e.g., Hindi)

- Unigram and Bigram Taggers – for training the POS model

4. Methodology

4.1 Tokenization:

Tokenization is done using the `indic_tokenize` module, which is suited for languages like Hindi, Marathi, Tamil, etc.

4.2 Training Data:

A small manually created POS-tagged corpus is used for training. Each sentence is a list of tuples in the form (word, tag).

4.3 Tagger Design:

- A DefaultTagger is used to assign 'NN' (noun) if no better guess is available.

- A UnigramTagger learns from individual word-tag pairs.

- A BigramTagger considers the current word and the previous word for context.

- The taggers are chained using a backoff strategy to increase accuracy.

5. Implementation

```
# Sample training data

train_data = [

   [('मैं', 'PRP'), ('स्कूल', 'NN'), ('जा', 'VM'), ('रहा', 'VAUX'), ('हूँ', 'VAUX')],

   [('वह', 'PRP'), ('घर', 'NN'), ('गया', 'VM')],

   [('हम', 'PRP'), ('खेल', 'NN'), ('रहे', 'VAUX'), ('थे', 'VAUX')],

]

# Train the tagger

default_tagger = nltk.DefaultTagger('NN')

unigram_tagger = nltk.UnigramTagger(train_data, backoff=default_tagger)

bigram_tagger = nltk.BigramTagger(train_data, backoff=unigram_tagger)

# Apply to test sentence

sentence = "मैं स्कूल जा रहा हूँ।"

tokens = list(indic_tokenize.trivial_tokenize(sentence, lang='hi'))

tagged = bigram_tagger.tag(tokens)
```

6. Sample Output

Tokenized sentence: ['मैं', 'स्कूल', 'जा', 'रहा', 'हूँ', '।']

POS Tagged Sentence:

मैं --> PRP

स्कूल --> NN

जा --> VM

रहा --> VAUX

हूँ --> VAUX

। --> NN

7. Conclusion

This mini project demonstrates a simple but effective POS tagging pipeline for Indian languages using rule-based and statistical taggers. Even with limited data, Unigram and Bigram taggers backed by appropriate tokenization give promising results.

8. Future Scope

- Use large annotated corpora like the Hindi Dependency Treebank.

- Implement CRF or Transformer-based taggers (e.g., BERT-based models).

- Extend support to more Indian languages with shared linguistic resources.

- Integrate with spaCy or Stanza for improved accuracy and pre-trained models.