Department of Computer Engineering

Zeal Education Society’s

Zeal College Of Engineering And Research

Narhe, Pune-411041

BE Computer Engineering

(Academic Year: 2022-2023)

Mini-Project

Topic

POS Taggers For Indian Languages using NLP

GUIDED BY

Prof. Dipali Pawar

Submitted By

Ajitkumar Vishwakarma Sharma

B212048

BE Division: B

CERTIFICATE

This is to certify **Mr. Ajitkumar V. Sharma**(Exam Seat No. B212048) of class B.E COMP; have successfully completed their mini project work on "POS Taggers for Indian Languages using NLP" at **ZEAL College of Engineering and Research, Pune** the partial fulfillment of the Graduate Degree course in B.E at the department of **Computer Engineering**, in the academic Year 2022-2023 Semester – has prescribed by the **Savitribai Phule Pune University.**

Prof. Dipali Pawar	Prof. A. V. Mote
Guide	Head of the Department
	(Department of Computer Engineering)

Title:

POS Tagger for Indian Languages using NLP.

Requirements:

nltk library
punkt
corpus
Indian library

Theory:

Natural language Processing:

Natural Language Processing (NLP) is a field of computer science and artificial intelligence that deals with the interaction between computers and humans in natural language. NLP is concerned with how to make computers understand, interpret, and generate human language, including text and speech.

NLP involves a range of techniques and approaches that enable computers to understand natural language, including:

1. Tokenization: This involves breaking down a text into smaller pieces, such as words or sentences. This is a necessary step for further analysis of the text.
2. Part-of-speech (POS) tagging: This is the process of identifying the grammatical parts of a sentence, such as nouns, verbs, adjectives, and adverbs. POS tagging is used to understand the meaning of the text and to build a more accurate representation of the text.
3. Named entity recognition (NER): This is the process of identifying and classifying named entities, such as people, organizations, and locations, within a text.
4. Sentiment analysis: This involves determining the emotional tone of a text, such as positive, negative, or neutral. Sentiment analysis can be used to analyze social media data or customer reviews, for example.
5. Machine translation: This involves translating text from one language to another. Machine translation is used to enable communication across language barriers.
6. Text summarization: This involves generating a shorter summary of a longer text. Text summarization can be used to quickly understand the main points of a longer document.
7. Question answering: This involves generating an answer to a natural language question, such as a question asked by a user to a virtual assistant.

NLP techniques are used in a wide range of applications, including language translation, chatbots, virtual assistants, sentiment analysis, and search engines. NLP is a rapidly developing field with many challenges, such as dealing with ambiguity and understanding idiomatic expressions. However, advances in NLP are making it possible for computers to better understand and communicate with humans in natural language.

POS Tagging for Indian Languages:

Part-of-speech (POS) tagging for Indian languages involves identifying and labeling the grammatical parts of a sentence, such as nouns, verbs, adjectives, and adverbs. POS tagging is a crucial step in natural language processing for Indian languages, as it helps to extract meaningful information from the text and enables further analysis.

POS tagging for Indian languages is challenging due to several reasons such as complex grammar, morphological richness, and variability in word order. However, there are several approaches and techniques that have been developed to address these challenges.

1. Rule-based approach: In this approach, a set of rules are defined based on the language grammar and morphology to assign POS tags to words. These rules are created manually by linguistic experts, and they can be language-specific. However, this approach requires a lot of time and effort to create and maintain the rules, and it may not be suitable for languages with complex grammar and morphological richness.
2. Statistical approach: This approach involves training a machine learning model on a large corpus of labeled data to predict the POS tags of words in a sentence. The model learns the patterns and associations between words and their corresponding POS tags. This approach is more scalable and can handle a wide range of languages and domains. However, it requires a large amount of labeled data for training, which may not be available for all languages.
3. Hybrid approach: This approach combines the rule-based and statistical approaches to achieve better accuracy and coverage. The rules are used to handle specific cases and exceptions, while the statistical model is used to handle general cases. This approach can achieve high accuracy and coverage while reducing the amount of manual effort required to create and maintain the rules.

There are several tools and libraries available for POS tagging for Indian languages, such as NLTK, SpaCy, and Stanford NLP. These tools and libraries support different approaches and techniques for POS tagging and can be used for different languages and domains.

In summary, POS tagging for Indian languages is a challenging task due to complex grammar and morphological richness. However, there are several approaches and techniques that have been developed to address these challenges, including rule-based, statistical, and hybrid approaches. These approaches can be used in combination with different tools and libraries to achieve high accuracy and coverage for different languages and domains.

Libraries used:

1. `nltk` - This is the main library for natural language processing in Python. It provides a wide range of tools and resources for working with text data, including tokenization, POS tagging, and text classification.
2. `word_tokenize` - This is a function from the `nltk.tokenize` module that is used to tokenize a text into individual words. Tokenizationis the process of breaking down a text into smaller units, such as words or sentences, in order to process them more easily.
3. `tnt` - This is a module from the `nltk.tag` package that provides an implementation of the Trigram Tagger (TnT) algorithm for POS tagging. TnT is a statistical algorithm that uses a combination of word and context features to predict the POS tag of a word.
4. `indian` - This is a corpus from the `nltk.corpus` module that contains several Indian language datasets, including the Indian POS tagged corpus. The Indian POS tagged corpus contains a large number of sentences annotated with their corresponding POS tags for several Indian languages, including Hindi, Bengali, and Telugu.

Together, these libraries provide a comprehensive set of tools for performing POS tagging on Indian language text data. The `word_tokenize` function is used to tokenize the text into individual words, which are then passed to the `tnt` tagger to assign POS tags to each word. The `indian` corpus can be used to train and evaluate the accuracy of the POS tagger on Indian language data.

In [1]:

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

# Download the Indian language corpora
#nltk.download('punkt')
#nltk.download('indian')

# Load the Hindi and Marathi POS tagged corpora
hindi_tagged = indian.tagged_sents('hindi.pos')
marathi_tagged = indian.tagged_sents('marathi.pos')

# Train the TnT taggers on the Hindi and Marathi POS tagged corpora
tnt_pos_tagger_hindi = tnt.TnT()
tnt_pos_tagger_hindi.train(hindi_tagged)

tnt_pos_tagger_marathi = tnt.TnT()
tnt_pos_tagger_marathi.train(marathi_tagged)

while True:
    # Ask the user for the language and input sentence
    language = input("Enter the language (hindi/marathi) or type 'exit' to qu

    if language == "exit":
        break

    input_sentence = input("Enter the input sentence: ")

    # Tokenize the input sentence
    input_tokens = word_tokenize(input_sentence)

    # Define a switch case construct to select the appropriate trained TnT ta
    def switch_case(language):
        switcher = {
            "hindi": tnt_pos_tagger_hindi,
            "marathi": tnt_pos_tagger_marathi
        }
        return switcher.get(language, "Invalid language specified!")

    # Get the appropriate trained TnT tagger based on the input language
    tagger = switch_case(language)

    # If the input language is invalid, print an error message and continue
    if tagger == "Invalid language specified!":
        print(tagger)
        print()
        continue

    # Tag the tokens with the selected TnT tagger
    pos_tags = tagger.tag(input_tokens)

    # Print the tagged tokens
    print(pos_tags)
    print()
```

```
Enter the language (hindi/marathi) or type 'exit' to quit: marathi
Enter the input sentence: हा खाद्य आणि वाटप परिपूर्ण आहे.
[('हा', 'DEM'), ('खाद्य', 'Unk'), ('आणि', 'CC'), ('वाटप', 'Unk'), ('परिपूर्ण', 'U
nk'), ('आहे', 'VAUX'), ('.', 'SYM')]

Enter the language (hindi/marathi) or type 'exit' to quit: hindi
Enter the input sentence: वह बचपन से मेरा सबसे अच्छा दोस्त है।
[('वह', 'PRP'), ('बचपन', 'Unk'), ('से', 'PREP'), ('मेरा', 'PRP'), ('सबसे', 'IN
TF'), ('अच्छा', 'JJ'), ('दोस्त', 'Unk'), ('है।', 'Unk')]

Enter the language (hindi/marathi) or type 'exit' to quit: exit
```

here are some sample inputs in Hindi and Marathi

Hindi:

- मैं आज काम पर था।
- वह एक अच्छा वकील है।
- ये फ़ोन बहुत अच्छा है।
- वह बचपन से मेरा सबसे अच्छा दोस्त है।
- यह एक सुंदर दिन है।

Marathi:

- तो सर्वोत्कृष्ट विद्यार्थी आहे.
- तो एक छान अभिनेता आहे.
- त्याचं काम खूप सुंदर आहे.
- हे पुस्तक तुमच्या प्रत्येकासाठी उपयुक्त आहे.
- हा खाद्य आणि वाटप परिपूर्ण आहे.