

# **Machine Learning approach to POS Tagging in Bengali Language**

## **PROJECT REPORT**

*Submitted by*

*Rajdeep Das (12619010037)*

*Arghyadeep Banerjee (12619010011)*

*Soham Chakraborty (12619010051)*

*Tanmay Guchhait (12619010056)*

*Debabrata Maity (12619010016)*

*Alik Sarkar (12619010004)*

*Sanju Manna (12619010043)*

*inpartial fulfillment for 5<sup>th</sup> Semester Minor Project*

**OF**

**MASTER OF COMPUTER APPLICATIONS**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA**

**MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY, WEST BENGAL**

**MARCH, 2021**

## **ACKNOWLEDGEMENT**

We are highly indebted to our project guide Prof. *Sandipan Ganguly* for guiding us and providing constant supervision and necessary information regarding the project and also for support in completing the project. We would also like to thank Mr. *Akash Modak* for guiding us along the way.

Rajdeep Das

Arghyadeep Banerjee

Soham Chakraborty

Tanmay Guichhait

Debabrata Maity

Alik Sarkar

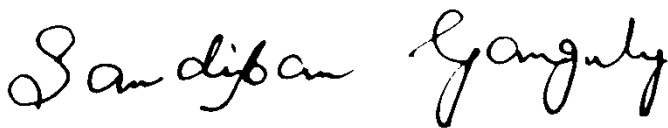
Sanju Manna

**DEPARTMENT OF COMPUTER APPLICATION**  
**HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA**



**BONAFIDE CERTIFICATE**

Certified that this project report **“Machine Learning Approach to POS Tagging in Bengali”** is the bona fide work of Rajdeep Das, Arghyadeep Banerjee, Soham Chakraborty, Tanmay Guchhait, Debabrata Maity, Alik Sarkar, Sanju Manna, students of MCA 5<sup>th</sup> Semester of Heritage Institute of Technology, Kolkata, who carried out the project work under my supervision.

A handwritten signature in black ink that reads "Sandipan Ganguly". The signature is written in a cursive style with a horizontal line underneath.

---

**Prof. Sandipan Ganguly**  
Mentor  
Assistant Professor, Department of Computer  
Applications  
Heritage Institute of Technology, Kolkata

### Declaration by Student

This is to declare that this report has been written by us. No part of the report is plagiarized from other sources. All information included from other sources has been duly acknowledged. We aver that if any part of the report is found to be plagiarized, we'll take full responsibility for it.

*Rajdeep Das*

Rajdeep Das (12619010037)

*Arghyadeep Banerjee*

Arghyadeep Banerjee (12619010011)

*Soham Chakraborty*

Soham Chakraborty (12619010051)

*Tanmay Guchhait*

Tanmay Guchhait (12619010056)

*Debabrata Maity*

Debabrata Maity (126169010016)

*Alik Sarkar*

Alik Sarkar (12619010004)

*Sanju Manna*

Sanju Manna (12619010043)

## Table of Contents:

Content	Page Number
1. Abstract & Introduction	6
2. POS Tagging on English Text	6-7
3. Bengali Corpus POS Tagging: Reasons behind Bengali POS Tagging Challenges of Bengali POS Tagging	7-8
4. BNLP Toolkit	9
5. Data Collection & Processing	10-13
6. Features and Approaches: · Rule-based approach · Conditional Random Field based approach	13-16
4. Sentence-piece Tokenizer	16
5. Confusion Matrices and Performance Analysis	17-18
7. Code-snippets and explanation	18-21
8. Accuracy	22-23
9. References	24-25

## **ABSTRACT:**

Part-of-speech (POS) tagging is a popular Natural Language Processing technique which refers to categorizing words in a text (corpus) in correspondence with a particular part of speech, depending on the definition of the word and its context. It consists of assigning to each word of a text the proper morphosyntactic tag in its context of appearance. It is very useful for a number of NLP applications: as a pre-processing step to syntactic parsing, in information extraction and retrieval (e.g. document classification in internet searchers), text to speech systems, corpus linguistics, etc.

POS tagging techniques, including eight stochastic based methods and eight transformation-based methods.

A comparative analysis of the tagging methods is performed with accuracy measures. CRF shows the highest accuracy of 80% among all the tagging techniques.

## **INTRODUCTION:**

Natural language processing is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language.

Now Natural language processing (NLP) tools have sparked a great deal of interest due to rapid improvements in information and communications technologies.

As a result, many different NLP tools are being produced. However, there are many challenges for developing efficient and effective NLP tools that accurately process natural languages. One such tool is part of speech (POS) tagging.

## **POS Tagging on English Text:**

A set of all POS tags used in a corpus is called a tagset. Tagsets for different languages are typically different. They can be completely different for unrelated languages and very similar for similar languages, but this is not always the rule. Tagsets can also go to a different level of detail. Basic tagsets may only include tags for the most common parts of speech (N for noun, V for verb, A for adjective etc.). In traditional POS Tagging, English is the most suitable to mostly used easy

to tagging corpus for POS Tagging. Because all the POS tags like nouns, pronouns, verbs, adverbs, adjectives are more compatible for tagging English dataset than any other language. For example,

```
In [6]: import nltk
        from nltk.tokenize import word_tokenize
        text = word_tokenize("This is a sample POS Tagging Testing on English Text.")
        nltk.pos_tag(text)

Out[6]: [('This', 'DT'),
         ('is', 'VBZ'),
         ('a', 'DT'),
         ('sample', 'JJ'),
         ('POS', 'NNP'),
         ('Tagging', 'NNP'),
         ('Testing', 'NNP'),
         ('on', 'IN'),
         ('English', 'NNP'),
         ('Text', 'NNP'),
         ('.', '.')]

```

In the above example, we have used NLTK (Natural Language Toolkit) to tag a English raw text and the outcome is as shown in the above picture.

## Bengali Corpus POS Tagging:

Bengali is the second most widely spoken of the 22 scheduled languages of India. With approximately 300 million native speakers and another 37 million as second language speakers, Bengali is the fifth most-spoken native language and the sixth most spoken language by total number of speakers in the world. Therefore, in addition to expanding the field of Bengali language research for this large number of people, it is necessary to enhance modern artificial intelligence-based technology.

Bengali is a highly inflectional language with **more than 160 different inflected forms for verbs and 36 different forms for nouns, and 24 different forms for pronouns**. Developing an efficient POS tagger is a challenging task for resource-scarce languages like Bengali. Like in English language, the Part of speeches is

noun, verb, adjective, adverb, pronoun, preposition, conjunction, interjection etc which is similar in the Bengali language.

In Bengali we can also tokenize & classify words into their corresponding Parts of Speech tags. Such classifiers help computers in understanding Bengali language, used in different regional real life applications.

## Challenges in Bengali POS Tagging:

- The main challenge of POS tagging comes due to the ambiguity in language. In Bengali, a large percentage of the words in a corpus are ambiguous.
- Bengali is a morphologically rich language. It is a highly agglutinative language, because of which the vocabulary size grows of a high rate with increase in the size of the corpus.
- Bengali is a relatively free word order language in comparison with European Languages.

As an example, we can consider the English Sentence:

I eat rice (PRP VB NN)

We have the following possible Bengali equivalents of the sentence:

- **Ami vat khai (I rice eat) (PPR NC VM)**
- **Ami khai vat (I eat rice) (PPR VM NC)**
- **Vat ami khai (rice I eat) (NC PPR VM)**
- **Vat khai ami (rice eat I ) (NC VM PPR)**
- **Khai ami vat (Eat I rice) (VM PPR NC)**
- **Khai vat ami (Eat rice I) (VM NC PPR)**

Hence, Part of Speech tagging using Linguistic rules is a difficult problem for such kinds of languages.

```
In [4]: from bnlp import POS
bn_pos = POS()
model_path = "bn_pos.pkl"
text = "আমি ভাত খাই।"
res = bn_pos.tag(model_path, text)
print(res)

[('আমি', 'PPR'), ('ভাত', 'NC'), ('খাই', 'VM'), ('।', 'PU')]
```



# BNLP Toolkit:

BNLP is an open source language processing toolkit for Bengali language consisting with tokenization, word embedding, POS tagging, NER tagging facilities. BNLP provides pre-trained model with high accuracy to do model based tokenization, embedding, POS tagging, NER tagging task for Bengali language. BNLP pre-trained model achieves significant results in Bengali text tokenization, word embedding, POS tagging and NER tagging task.

BNLP is providing different tokenization method to tokenize Bengali text efficiently

- BNLP Provides different types of embedding method using their pre-trained models to embed Bengali word it also provides an option to train and embedding model from scratch.
- It provides hands on start option for pos tagging or Name Entity Recognition (NER) tagging of Bengali sentences and also provides an option for training Conditional Random Field based (CRF) approach to pos tagger or NER tagger model from scratch.
- BNLP also provides some utility methods like to remove stop-words from Bengali text, to get Bengali letters list or punctuation list.
- BNLP libraries have a permissive MIT license. BNLP is easy to install via pip or by cloning repository with any python projects.

## Installation

pypi package installer(python 3.6, 3.7, 3.8 tested okay)

`pip install bnlp_toolkit`

or Upgrade

`pip install -U bnlp_toolkit`

## DATA COLLECTION AND PROCESSING:

Each word has its own lexical term written underneath, however, having to constantly write out these full terms when we perform text analysis can very quickly become cumbersome — especially as the size of the corpus grows. Thence, we use a short representation referred to as “tags” to represent the categories. Part-of-speech tags describe the characteristic structure of lexical terms within a sentence or text; therefore, we can use them for making assumptions about semantics. When we perform POS tagging, it’s often the case that our tagger will encounter words that were not within the vocabulary that was used. Consequently, augmenting your dataset to include unknown word tokens will aid the tagger in selecting appropriate tags for those words. For POS tagging we used NLTR 8 datasets which contains total of 2997 sentences.

We split that datasets into 2247 train and 750 tests set and train our POS tagging model. BNLTP toolkit has been used to tag Bengali words & punctuations using training dataset of NLTR with 80% accuracy & has Bengali CRF NER Tagging which was trained with this data with 90% accuracy. We divided data into 75% train and 25% test. Our evaluation result for POS tagging model is 80.75 F1.

---

	<b>Sentences</b>	<b>Train</b>	<b>Test</b>
<b>POS</b>	2997	2247	750

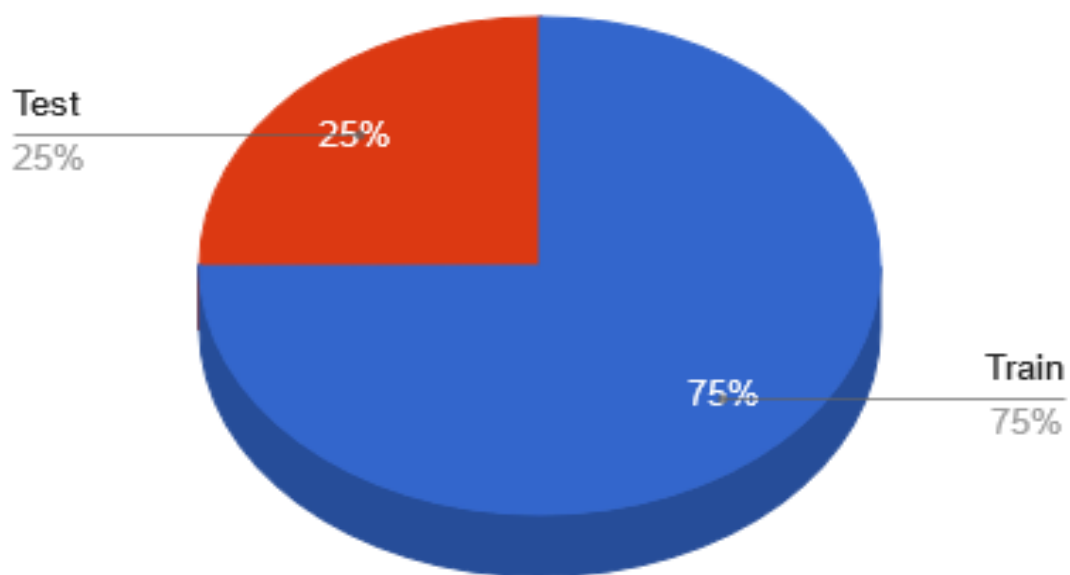
Statistics of POS Dataset from NLTR

---

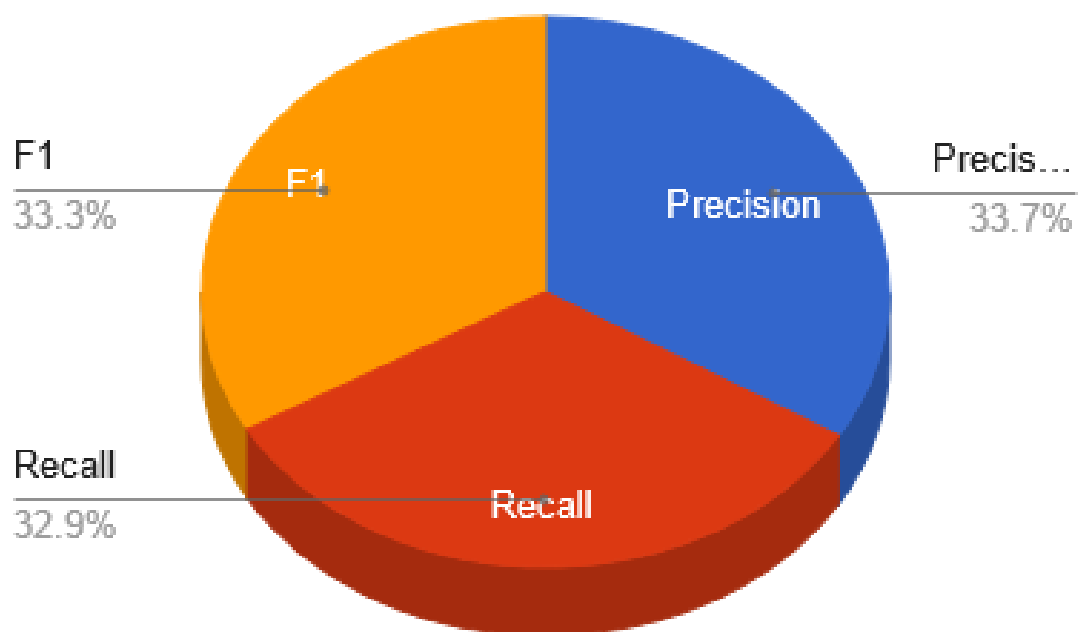
	<b>Precision</b>	<b>Recall</b>	<b>F1</b>
<b>POS</b>	81.74	79.78	80.75

Evaluation result of POS

### BNLP POS Tagging using NLTR Dataset

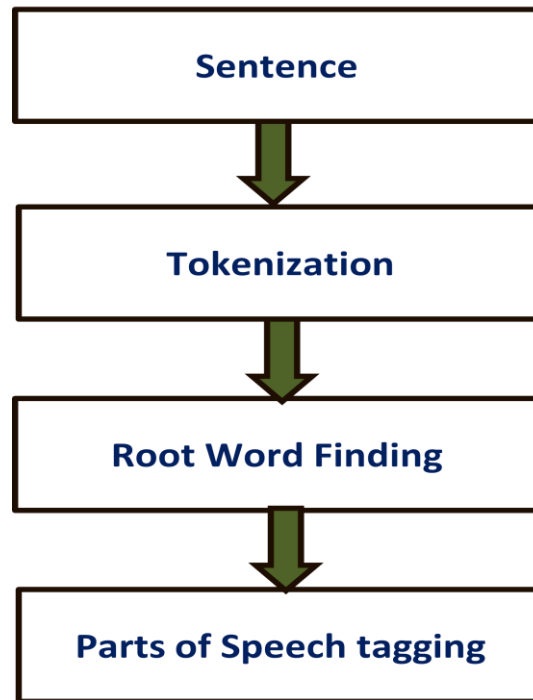


### Evaluated result of BNLP POS Tagging using NLTR Dataset



## Preprocessing on Raw dataset

### Raw Text



### Sample Testing on Preprocessing

Raw text = "আমি ভাত খাই।"

Taking above sentence to apply POS Tagging.

Sentence= "আমি ভাত খাই।"

Tokenization:

```
In [3]: from bnlp import BasicTokenizer
basic_t = BasicTokenizer()
raw_text = "আমি ভাত খাই।"
tokens = basic_t.tokenize(raw_text)
print(tokens)

['আমি', 'ভাত', 'খাই', '।']
```

## Parts of Speech Tagging

```
In [4]: from bnlp import POS
bn_pos = POS()
model_path = "bn_pos.pkl"
text = "আমি ভাত খাই।"
res = bn_pos.tag(model_path, text)
print(res)

[('আমি', 'PPR'), ('ভাত', 'NC'), ('খাই', 'VM'), ('।', 'PU')]
```

So here, we have taken a raw text as a sentence to apply first POS Tagging where first we have applied tokenization using BasicTokenizer(). After tokenization, we have applied POS() for POS Tagging.

```
[('আমি', 'PPR'), ('ভাত', 'NC'), ('খাই', 'VM'), ('।', 'PU')]
```

## Features and Approaches:

### Rule-based approach:

One of the oldest techniques of tagging is rule-based POS tagging. Rule-based taggers use a dictionary (i.e. it can store a number of words) or lexicon for getting possible tags for tagging each word. If the word has more than one possible tag, then rule-based taggers use hand-written rules to identify the correct tag. For example, suppose if the preceding word of a word is an article then the word must be a noun. All such kind of information in rule-based POS tagging is coded in the form of rules. These rules may be either –

- Context-pattern rules
- Or, as Regular expression compiled into finite-state automata, intersected with lexically ambiguous sentence representation.

```
In [4]: import nltk
from nltk.tokenize import word_tokenize
text = word_tokenize("Rabindranath Tagore, Bengali Rabindranāth Thākur, (born May 7, 1861, Calcutta [now Kolkata], India-died")
nltk.pos_tag(text)

Out[4]: [('Rabindranath', 'NNP'),
('Tagore', 'NNP'),
(',', ','),
('Bengali', 'NNP'),
('Rabindranāth', 'NNP'),
('Thākur', 'NNP'),
(',', ','),
('(', '('),
('born', 'VBN'),
('May', 'NNP'),
('7', 'CD'),
(',', ','),
('1861', 'CD'),
(',', ','),
('Calcutta', 'NNP'),
('[' , 'NNP'),
('now', 'RB'),
('Kolkata', 'NNP'),
(')', 'NNP'),
('India-died', 'NNP')]
```

Now we have test this same approach on Bengali Text POS Tagging using BNLPToolkit.

BNLP provides three different tokenization options to tokenize Bengali text. Under rule based approach BNLP provides “Basic Tokenizer”, a “Punctuation Splitting Tokenizer” and “NLTK Tokenizer”.

1. Using Basic Tokenizer we get:

```
In [2]: from bnlp import BasicTokenizer
basic_t = BasicTokenizer()
raw_text = "আমি বাংলায় গান গাই।"
tokens = basic_t.tokenize(raw_text)
print(tokens)

['আমি', 'বাংলায়', 'গান', 'গাই', '.']
```

Here, under rule based approach, we have tokenized each word & punctuation using ‘BasicTokenizer’ from BNLP on the following raw text.

## 2. Using NLTK Tokenizer we get:

```
In [3]: from nltk import NLTKTokenizer

nltk = NLTKTokenizer()

text = "আমি ভাত খাই। সে বাজারে যায়। তিনি কি সত্যিই ভালো মানুষ?"

word_tokens = nltk.word_tokenize(text)
sentence_tokens = nltk.sentence_tokenize(text)
print(word_tokens)
print(sentence_tokens)

['আমি', 'ভাত', 'খাই', '.', 'সে', 'বাজারে', 'যায়', '.', 'তিনি', 'কি', 'সত্যিই', 'ভালো', 'মানুষ', '?']
['আমি ভাত খাই।', 'সে বাজারে যায়।', 'তিনি কি সত্যিই ভালো মানুষ?']
```

Under Rule based approach, we have tokenized the above raw text using 'NLTKTokenizer' in two phases. One is Word Tokenization and another is Sentence Tokenizer. In Word Tokenization, we have tokenized each word from each sentence and in sentence tokenization; we have tokenized 3 sentences into 3 tokens.

## Conditional Random Field based approach:

A CRF is a sequence modeling algorithm which is used to identify entities or patterns in text, such as POS tags. This model not only assumes that features are dependent on each other, but also considers future observations while learning a pattern. In terms of performance, it is considered to be the best method for entity recognition.

- CRF is a discriminative probabilistic classifier
- In CRF based models, the input is:
  - A set of feature derived from the input sequence.
  - Weights associated with the features.
  - Previous label
- Our task is to predict the correct label

The feature functions express certain characteristics of the sequence. Example:  
the tag sequence noun-> verb -> adjective

```
In [5]: from bnlp import POS
bn_pos = POS()
model_path = "bn_pos.pkl"
text = "আমি বাংলায় গান গাই।"
res = bn_pos.tag(model_path, text)
print(res)

[('আমি', 'PPR'), ('বাংলায়', 'NP'), ('গান', 'NC'), ('গাই', 'NC'), ('।', 'PU')]
```

## SentencePiece Tokenizer:

SentencePiece is an unsupervised text tokenizer and detokenizer mainly for Neural Network-based text generation systems where the vocabulary size is predetermined prior to the neural model training. SentencePiece implements subword units (e.g., byte-pair-encoding (BPE)).

Here we have also SentencePiece Tokenizer on Bengali Raw Text as follows:

```
In [6]: from bnlp import SentencepieceTokenizer

bsp = SentencepieceTokenizer()
model_path = "bn_spm.model"
input_text = "আমি ভাত খাই। সে বাজারে যায়।"
tokens = bsp.tokenize(model_path, input_text)
print(tokens)

['_আমি', '_ভাত', '_খাই', '।', '_সে', '_বাজারে', '_যায়', '।']
```



# Confusion Matrices:

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one.

By definition a confusion matrix  $C$  is such that  $C(i, j)$  is equal to the number of observations known to be in group  $i$ , and predicted to be in group  $j$ .

Thus in binary classification, the count of

- True negatives is  $C(0,0)$  - TN
- False negatives is  $C(1,0)$  - FN
- True positives is  $C(1,1)$  - TP
- False positives is  $C(0,1)$  - FP

In this project, while using pre-trained model of Bengali POS Tagger, we have retrieved our values from Precision and Recall from following calculations:

Precision= 81.74:  $TP/(TP+FP)$

Recall=79.78:  $TP/(TP+FN)$

```
In [5]: from bnlp import POS
bn_pos = POS()
model_path = "bn_pos.pkl"
text = "আমি বাংলায় গান গাই।"
res = bn_pos.tag(model_path, text)
print(res)

[('আমি', 'PPR'), ('বাংলায়', 'NP'), ('গান', 'NC'), ('গাই', 'NC'), ('।', 'PU')]
```

In the above image, the term “গাই” is expected to be tagged as VM (Verb) but it is tagged as NC (Noun). That means, it can be considered as FALSE POSITIVE.

```
from bnlp import POS
bn_pos = POS()
model_path = "bn_pos.pkl"
text = "আমি ভাত খাই।"
res = bn_pos.tag(model_path, text)
print(res)
```

Unlike the previous one, here the term “খাই” is correctly tagged as VM (Verb). So it is TRUE POSITIVE.

## Code-Snippet

We have collected corpus from NLTR on which we have applied all our above mentioned POS Tagging approaches using BNLTP Toolkit.

### 1. Code-snippet 1

```
In [4]: from bnlp import BasicTokenizer
basic_t = BasicTokenizer()
raw_text = "বাংলা সাহিত্যের নবজাগরণের পথিকৃৎ-কমী বঙ্কিমচন্দ্র চট্টোপাধ্যায়। বঙ্কিমচন্দ্রের জন্ম ১৮৩৮ সালের ২৬ শে জুন, অধুনা চব্বিশ পরগণা জেলার ।
tokens = basic_t.tokenize(raw_text)
print(tokens)
```

['বাংলা', 'সাহিত্যের', 'নবজাগরণের', 'পথিকৃৎ', '-', 'কমী', 'বঙ্কিমচন্দ্র', 'চট্টোপাধ্যায়', '।', 'বঙ্কিমচন্দ্রের', 'জন্ম', '১৮৩৮', 'সালের', '২৬', 'শে', 'জুন', 'অধুনা', 'চব্বিশ', 'পরগণা', 'জেলার', 'অন্তর্গত', 'নৈহাটির', 'কাঁঠালপাড়া', 'গ্রামে', '।', 'বাবা', 'যাদবচন্দ্র', 'চট্টোপাধ্যায়', 'ছিলেন', 'মেদিনীপুরের', 'কলেঙ্কুর', '।', 'নিয়মমাফিক', 'পড়াশোনা', 'শুরু', 'বাবার', 'কর্মস্থল', 'মেদিনীপুর', 'জেলার', 'এক', 'ইংরেজি', 'স্কুলে', '।', 'পরে', 'কাঁঠালপাড়ায়', 'ফিরে', 'হুগলি', 'কলেজে', '।', '১৮৫৬', 'সালে', 'বঙ্কিমচন্দ্র', 'আইন', 'পড়বার', 'জন্ম', 'প্রেসিডেন্সি', 'কলেজে', 'ভর্তি', 'হন', 'এবং', '১৮৫৭তে', 'সেখান', 'থেকে', 'প্রথম', 'বিভাগে', 'এন্ট্রান্স', 'পরীক্ষা', 'পাশ', 'করেন', '।', '১৮৫৮', 'সালে', 'সদ্য', 'প্রতিষ্ঠিত', 'কোলকাতা', 'বিশ্ববিদ্যালয়ের', 'প্রথম', 'বি', 'এ', 'পরীক্ষায়', 'বঙ্কিমচন্দ্র', 'দ্বিতীয়', 'বিভাগে', 'প্রথম', 'স্থান', 'অধিকার', 'করেন', '।', 'আইন', 'পড়া', 'শেষ', 'হওয়ার', 'আগেই', 'যশোরের', 'ডেপুটি', 'ম্যাজিষ্ট্রেট', 'ও', 'ডেপুটি', 'কলেঙ্কুরের', 'চাকরি', 'পান', '।']

**Explanation:**

In the above code snippet, we have applied `BasicTokenizer()`.

First we have imported BasicTokenizer function from BNL.

Taking a variable `basic_t` we have assigned the `BasicTokenizer` function inside it.

Taking a raw text on Bankim Chandra Chattopadhyay's lifestory in raw\_text variable,

Stored the result in tokens variable using `basic_t.tokenize()` function with `raw_text` variable.

## 2. Code-snippet 2

```
In [4]: from bnlp import NLTKTokenizer

bnltk = NLTKTokenizer()

text = "বাংলা সাহিত্যের নবজাগরণের পথিকৃৎ-কর্মী বঙ্কিমচন্দ্র চট্টোপাধ্যায়। বঙ্কিমচন্দ্রের জন্ম ১৮৩৮ সালের ২৬ শে জুন, অধুনা চব্বিশ পরগণা জেলার অন্তর্গত নৈহাটির কাঁঠালপাড়া গ্রামে। 'বাবা' যাদবচন্দ্র চট্টোপাধ্যায় ছিলেন মেদিনীপুরের কলেঙ্কুর। 'নিয়মমাফিক' পড়াশোনা শুরু বাবার কর্মস্থল মেদিনীপুর জেলার এক ইংরেজি স্কুলে। 'পরে' কাঁঠালপাড়ায় ফিরে 'হুগলি' কলেজে। '১৮৫৬' সালে 'বঙ্কিমচন্দ্র' আইন পড়বার জন্য 'প্রেসিডেন্সি' কলেজে 'ভর্তি' হন 'এবং' '১৮৫৭'তে 'সেখান' 'থেকে' 'প্রথম' 'বিভাগে' 'এন্ট্রান্স' পরীক্ষা পাশ করেন। '১৮৫৮' সালে 'সদ্য' 'প্রতিষ্ঠিত' 'কোলকাতা' বিশ্ববিদ্যালয়ের 'প্রথম' 'বি.এ. পরীক্ষায়' 'বঙ্কিমচন্দ্র' 'দ্বিতীয়' 'বিভাগে' 'প্রথম' স্থান 'অধিকার' 'করেন'। 'আইন' 'পড়া' শেষ 'হওয়ার' 'আগেই' 'যশোরের' 'ডেপুটি' 'ম্যাজিস্ট্রেট' ও 'ডেপুটি' 'কলেঙ্কুরের' 'চাকরি' 'পান'।

[ 'বাংলা', 'সাহিত্যের', 'নবজাগরণের', 'পথিকৃৎ-কর্মী', 'বঙ্কিমচন্দ্র', 'চট্টোপাধ্যায়', '.', 'বঙ্কিমচন্দ্রের', 'জন্ম', '১৮৩৮', 'সালের', '২৬', 'শে', 'জুন', 'অধুনা', 'চব্বিশ', 'পরগণা', 'জেলার', 'অন্তর্গত', 'নৈহাটির', 'কাঁঠালপাড়া', 'গ্রামে', 'বাবা', 'যাদবচন্দ্র', 'চট্টোপাধ্যায়', 'ছিলেন', 'মেদিনীপুরের', 'কলেঙ্কুর', 'নিয়মমাফিক', 'পড়াশোনা', 'শুরু', 'বাবার', 'কর্মস্থল', 'মেদিনীপুর', 'জেলার', 'এক', 'ইংরেজি', 'স্কুলে', 'পরে', 'কাঁঠালপাড়ায়', 'ফিরে', 'হুগলি', 'কলেজে', '১৮৫৬', 'সালে', 'বঙ্কিমচন্দ্র', 'আইন', 'পড়বার', 'জন্য', 'প্রেসিডেন্সি', 'কলেজে', 'ভর্তি', 'হন', 'এবং', '১৮৫৭তে', 'সেখান', 'থেকে', 'প্রথম', 'বিভাগে', 'এন্ট্রান্স', 'পরীক্ষা', 'পাশ', 'করেন', '১৮৫৮', 'সালে', 'সদ্য', 'প্রতিষ্ঠিত', 'কোলকাতা', 'বিশ্ববিদ্যালয়ের', 'প্রথম', 'বি.এ. পরীক্ষায়', 'বঙ্কিমচন্দ্র', 'দ্বিতীয়', 'বিভাগে', 'প্রথম', 'স্থান', 'অধিকার', 'করেন', 'আইন', 'পড়া', 'শেষ', 'হওয়ার', 'আগেই', 'যশোরের', 'ডেপুটি', 'ম্যাজিস্ট্রেট', 'ও', 'ডেপুটি', 'কলেঙ্কুরের', 'চাকরি', 'পান', '।']

[ 'বাংলা সাহিত্যের নবজাগরণের পথিকৃৎ-কর্মী বঙ্কিমচন্দ্র চট্টোপাধ্যায়।', 'বঙ্কিমচন্দ্রের জন্ম ১৮৩৮ সালের ২৬ শে জুন, অধুনা চব্বিশ পরগণা জেলার অন্তর্গত নৈহাটির কাঁঠালপাড়া গ্রামে।', 'বাবা যাদবচন্দ্র চট্টোপাধ্যায় ছিলেন মেদিনীপুরের কলেঙ্কুর।', 'নিয়মমাফিক পড়াশোনা শুরু বাবার কর্মস্থল মেদিনীপুর জেলার এক ইংরেজি স্কুলে।', 'পরে কাঁঠালপাড়ায় ফিরে হুগলি কলেজে।', '১৮৫৬ সালে বঙ্কিমচন্দ্র আইন পড়বার জন্য প্রেসিডেন্সি কলেজে ভর্তি হন এবং ১৮৫৭তে সেখান থেকে প্রথম বিভাগে এন্ট্রান্স পরীক্ষা পাশ করেন।', '১৮৫৮ সালে সদ্য প্রতিষ্ঠিত কোলকাতা বিশ্ববিদ্যালয়ের প্রথম বি.এ. পরীক্ষায় বঙ্কিমচন্দ্র দ্বিতীয় বিভাগে প্রথম স্থান অধিকার করেন।', 'আইন পড়া শেষ হওয়ার আগেই যশোরের ডেপুটি ম্যাজিস্ট্রেট ও ডেপুটি কলেঙ্কুরের চাকরি পান।']
```

### Explanation:

In the above code snippet, we have applied `NLTKTokenizer()`.

First we have imported `NLTKTokenizer` function from `BNLP`.

Taking a variable `bnltk`, we have assigned the `NLTKTokenizer` function inside it.

We have taken two variables, such as `word_tokens` and `sentence_tokens` and assigned two functions `bnltk.word_tokenize` and `bnltk.sentence_tokenize` respectively inside it with corresponding Bengali corpus.

Taking a raw text on Bankim Chandra Chattopadhyay's lifestory in the 'text' variable.

We have tokenized it in two phases. One is Word Tokenization and another is Sentence Tokenization.

### 3. Code-snippet 3

```
In [4]: from bnlp import POS
bn_pos = POS()
model_path = "bn_pos.pkl"
text = "বাংলা সাহিত্যের নবজাগরণের পথিকৃৎ-কমী বঙ্কিমচন্দ্র চট্টোপাধ্যায়। বঙ্কিমচন্দ্রের জন্ম ১৮৩৮ সালের ২৬ শে জুন, অধুনা চব্বিশ পরগণা জেলার অন্তর্গত  

print(res)
```

[('বাংলা', 'NP'), ('সাহিত্যের', 'NC'), ('নবজাগরণের', 'NC'), ('পথিকৃৎ', 'NC'), ('-', 'PU'), ('কমী', 'NP'), ('বঙ্কিমচন্দ্র', 'NP'), ('চট্টোপাধ্যায়', 'NP'), ('।', 'NP'), ('বঙ্কিমচন্দ্রের', 'NP'), ('জন্ম', 'NC'), ('১৮৩৮', 'RDF'), ('সালের', 'NC'), ('২৬', 'RDF'), ('শে', 'CX'), ('জুন', 'NP'), ('', 'PU'), ('অধুনা', 'NC'), ('চব্বিশ', 'JQ'), ('পরগণা', 'NC'), ('জেলার', 'NC'), ('অন্তর্গত', 'JQ'), ('নৈশ্চিহ্ন', 'NC'), ('কাঠালপাড়া', 'NC'), ('গ্রামে', 'NC'), ('', 'NP'), ('বাবা', 'NC'), ('যাদবচন্দ্র', 'NP'), ('চট্টোপাধ্যায়', 'NP'), ('ছিলেন', 'VM'), ('মেদিনীপুরের', 'NP'), ('কলেঙ্কর', 'NP'), ('', 'NP'), ('নিয়মমাফিক', 'JQ'), ('পড়াশোনা', 'NC'), ('শুরু', 'NC'), ('বাবার', 'NC'), ('কর্মস্থল', 'VM'), ('মেদিনীপুর', 'NP'), ('জেলার', 'NC'), ('এক', 'JQ'), ('ইংরেজি', 'NC'), ('কুলে', 'NC'), ('', 'NP'), ('পরে', 'NST'), ('কাঠালপাড়ায়', 'NC'), ('ফিরে', 'VM'), ('হুগলি', 'NP'), ('কলেজে', 'NP'), ('', 'NP'), ('১৮৫৬', 'RDF'), ('সালে', 'NC'), ('বঙ্কিমচন্দ্র', 'NP'), ('আইন', 'NC'), ('পড়বার', 'NV'), ('জন্য', 'PP'), ('প্রেসিডেন্সি', 'NC'), ('কলেজে', 'NC'), ('ভর্তি', 'NC'), ('হন', 'VM'), ('এবং', 'CCD'), ('১৮৫৭তে', 'NC'), ('সেখান', 'NC'), ('থেকে', 'PP'), ('প্রথম', 'JQ'), ('বিভাগে', 'NC'), ('এন্ড্রাস', 'NP'), ('পরীক্ষা', 'NC'), ('পাশ', 'NC'), ('করেন', 'VM'), ('', 'NP'), ('১৮৫৮', 'RDF'), ('সালে', 'NC'), ('সদ্য', 'NC'), ('প্রতিষ্ঠিত', 'JQ'), ('কোলকাতা', 'NC'), ('বিশ্ববিদ্যালয়ের', 'NC'), ('প্রথম', 'JQ'), ('বি', 'NC'), ('', 'NP'), ('এ', 'DAB'), ('', 'NP'), ('পরীক্ষায়', 'NC'), ('বঙ্কিমচন্দ্র', 'NP'), ('দ্বিতীয়', 'JQ'), ('বিভাগে', 'NC'), ('প্রথম', 'JQ'), ('স্থান', 'NC'), ('অধিকার', 'NC'), ('করেন', 'VM'), ('', 'NP'), ('আইন', 'NC'), ('পড়া', 'NV'), ('শেষ', 'NST'), ('হওয়ার', 'NV'), ('আগেই', 'NST'), ('যশোরের', 'NP'), ('ডেপুটি', 'NC'), ('ম্যাজিস্ট্রেট', 'NC'), ('ও', 'CCD'), ('ডেপুটি', 'NC'), ('কলেঙ্করের', 'NP'), ('চাকরি', 'NC'), ('পান', 'VM'), ('', 'PU')]

#### Explanation:

In the above code snippet, we have applied POS().

First we have imported POS function from BNLP.

Taking a variable bn\_pos, we have assigned the POS() function inside it.

We have taken a pre-trained model named 'bn\_pos.pkl' in model\_path variable to execute it on the corpus.

Taking a raw text on Bankim Chandra Chattopadhyay's lifestory in the 'text' variable, we have tagged each and every word into different Parts of Speech classifications.

Used bn\_pos.tag function with model path and text and stored the output in res variable.

## 4. Code-snippet 4

```
In [3]: from bnlp import SentencepieceTokenizer

bsp = SentencepieceTokenizer()
model_path = "bn_spm.model"
input_text = "বঙ্কিমচন্দ্র ও রবীন্দ্রনাথের পর বাংলা সাহিত্যের আকাশে উজ্জ্বলতম জ্যোতিষ্ক হলেন কথাসিদ্ধী শরৎচন্দ্র চট্টোপাধ্যায়, যিনি তাঁর সীমিত কালখণ্ডে  
tokens = bsp.tokenize(model_path, input_text)
print(tokens)
```

< [ 'বঙ্কিমচন্দ্র', 'ও', 'রবীন্দ্রনাথের', 'পর', 'বাংলা', 'সাহিত্যের', 'আকাশে', 'উজ্জ্বল', 'তম', 'জ্যোতিষ্ক', 'হলেন', 'কথা', 'সিদ্ধী', 'শরৎচন্দ্র', 'চট্টোপাধ্যায়', 'যিনি', 'তাঁর', 'সীমিত', 'কাল', 'খণ্ডে', 'ও', 'ভূমি', 'খণ্ডে', 'কে', 'রচনা', 'এ', 'অতিক্রম', 'করে', 'এক', 'যুগ', 'োত্তীর্ণ', 'মর্যাদায়', 'অধিষ্ঠিত', 'হয়ে', 'আছেন', 'বাঙালি', 'পাঠক', 'সমা', 'জে', 'তাঁর', 'কালজয়ী', 'খ্যাতি', 'দেশের', 'সীমা', 'কে', 'বঙ্কিমচন্দ্র', 'ও', 'রবীন্দ্রনাথের', 'পরি', 'লঙ্ঘন', 'ক', 'রে', 'বিশ্বের', 'বিভিন্ন', 'দেশে', 'বিস্তার', 'লাভ', 'করে', 'বিদেশি', 'পাঠকদের', 'মনকে', 'ও', 'জয়', 'করেছে', 'বাংলা', 'উপন্যাস', 'সাহিত্যে', 'শরৎচন্দ্র', 'এমন', 'একটি', 'নতুন', 'পথ', 'ধরে', 'অগ্রসর', 'হয়েছেন', 'যা', 'বাঙালি', 'কথাসাহিত্য', 'ের', 'পরিধি', 'কে', 'প্রসারিত', 'করে', 'দিয়ে', 'তার', 'মধ্যে', 'এনেছে', 'এক', 'অ', 'দৃ', 'ষ্ট', 'পূর্ব', 'বেচিরা', 'সংবেদনশীল', 'হৃদয়', 'ব্যাপক', 'জীবন', 'জি', 'জ্ঞা', 'সা', 'প্রখর', 'পর্যবেক্ষণ', 'শক্তি', 'সংস্কার', 'মুক্ত', 'স্থায়ী', 'মনো', 'ভঙ্গি', 'প্রভুতির', 'গুণে', 'শরৎ', 'সাহিত্য', 'লাভ', 'করেছে', 'এক', 'অনন্যসাধারণ', 'বিশিষ্ট', 'তা', 'যা', 'পরবর্তীকালের', 'বাঙালি', 'সাহিত্যের', 'গতিপ্রকৃতি', 'কে', 'অনেকাংশে', 'নিয়ন্ত্রিত', 'করেছে', 'শরৎচন্দ্র', 'ের', 'সমস্ত', 'উপন্যাস', 'ও', 'ছোট', 'গল্পগুলি', 'কে', 'প্রধানত', 'পারিবারিক', 'সামাজিক', 'ও', 'মনস্তত্ত্ব', 'মূলক', 'এই', 'তিন', 'শ্রেণীতে', 'বিভক্ত', 'করলেও', 'তার', 'অধিকাংশ', 'উপন্যাসের', 'কেন্দ্র', 'ভূমিতে', 'বিরাজমান', 'রয়েছে', 'বাঙালী', 'র', 'সমাজ', 'সম্পর্কে', 'এক', 'বিরাট', 'জিজ্ঞাসা', 'এ', 'বং', 'বাঙালির', 'মধ্যবিত্ত', 'শ্রেণীর', 'অন্তরঙ্গ', 'ও', 'বহি', 'রঙ্গ', 'জীবনের', 'রূপায়ণ', 'সমাজের', 'বাস্তব', 'অবস্থা', 'নর', 'নারীর', 'জীবন', 'ভঙ্গি', 'মা', 'ও', 'জীবন', 'বোধ', 'কে', 'নিয়ন্ত্রিত', 'করে', 'তাদের', 'মানস', 'লো', 'কে', 'যে', 'সুক্ষ্ম', 'জটিল', 'প্রতিক্রিয়ার', 'সৃষ্টি', 'করে', 'শরৎ', 'সাহিত্যে', 'আমরা', 'পাই', 'তারই', 'সার্থক', 'রূপায়ণ', 'বাঙালি', 'মধ্যবিত্ত', 'সমাজের', 'দুঃখ', 'বেদনা', 'র', 'এত', 'বড়', 'কাব্য', 'কার', 'ইতিপূর্বে', 'দে', 'খিনি', 'আমরা', 'া', 'মু', 'ঢ়', 'তায়', 'আচ্ছন্ন', 'সমাজব্যবস্থা', 'র', 'নিষ্কটর', 'শাসনে', 'লাঞ্ছিত', 'নর', 'নারীর', 'অশ্রু', 'সিক্ত', 'জীবন', 'কথা', 'অবলম্বন', 'করে', 'মানব', 'দর', 'দী', 'নিষ্কটর', 'শরৎচন্দ্র', 'গদ্য', 'বাহিত', 'যে', 'কতক', 'গুলি', 'উৎকৃষ্ট', 'ট্রাজেডি', 'রচনা', 'করেছেন', 'তাতে', 'বাঙালি', 'সমাজের', 'অতি', 'বিশ্ব', 'স্ত', 'ও', 'বহু', 'চিত্র', 'িত', 'এক', 'আলেখ্য', 'উন্মোচ', 'িত', 'হয়েছে', 'আমাদের', 'সামনে', '।']

### Explanation:

In the above code snippet, we have applied SentencepieceTokenizer().

First we have imported SentencepieceTokenizer function from BNLP.

Taking a variable bsp, we have assigned the SentencepieceTokenizer function inside it.

We have taken a pre-trained model names 'bn\_spm.model' in model\_path variable to execute it on the corpus.

Taking a raw text on Bankim Chandra Chattopadhyay's lifestory in input\_text variable, we have tokenized and detokenized into Sentence-pieces.

Used bsp.tokenize() function with model\_path and input\_text variables and stored the outcome in tokens variable.

## Accuracy:

Tagger: Unigram Tagger

0.647873865265 0.645268663814 0.634671890304 0.661375661376  
0.64314516129 0.666666666667 0.654210028382 0.625187406297  
0.657087189479 0.658976930792

Accuracy Score: 0.649446346367

Tagger: Unigram-Bigram Tagger

0.637048192771 0.652387986214 0.672032193159 0.663613231552  
0.662399241347 0.653611393693 0.654751131222 0.66061522945  
0.647086914995 0.654237288136

Accuracy Score: 0.655778280254

Tagger: Unigram-Tigram Tagger

0.66021708353 0.65369261477 0.656352555087 0.646687697161  
0.643867924528 0.645937358148 0.643706640238 0.678863745787  
0.652109911678 0.648975791434

Accuracy Score: 0.653041132236

Tagger: Unigram Bigram Trigram Tagger

0.658385093168 0.648886827458 0.670961347869 0.66004842615  
0.661089866157 0.660611854685 0.64859437751 0.639512195122  
0.658722592946 0.639906103286

Accuracy Score: 0.654671868435

Tagger: Affix based tagger

0.334864726901 0.355238095238 0.332535885167 0.346898263027  
0.335504885993 0.3415 0.354997538159 0.346314325452 0.341048653755  
0.337493759361

Accuracy Score: 0.342639613305

Tagger: Affix Unigram Bigram Tigram Tagger

0.773422562141 0.778386454183 0.766697163769 0.756005652379  
0.783231083845 0.766909975669 0.778246601031 0.757142857143  
0.773729626079 0.776486988848

Accuracy Score: 0.771025896509

Tagger: Brill Based Tagger with AUBT as the trainer Tagger 0.759141882009

0.774105930285 0.768367346939 0.770405727924 0.764204545455  
0.78997020854 0.782851344495 0.76412649041 0.770356816102  
0.758321273517

Accuracy Score: 0.770185156567

## References:

**Rural based Approach:**

[https://www.tutorialspoint.com/natural\\_language\\_processing/natural\\_language\\_processing\\_part\\_of\\_speech\\_tagging.htm](https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_part_of_speech_tagging.htm)  
[https://www.tutorialspoint.com/natural\\_language\\_processing/natural\\_language\\_processing\\_part\\_of\\_speech\\_tagging.htm](https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_part_of_speech_tagging.htm)

**Conditional Random Field:** <https://towardsdatascience.com/pos-tagging-using-crfs-ea430c5fb78b>

**Available Corpus: -**

**Bengali Stop-words Collected from:** <https://github.com/stopwords-iso/stopwords-bn>

**Bengali letters and vowel mark collected from**  
<https://github.com/MinhasKamal/BengaliDictionary/blob/master/BengaliCharacterCombinations.txt>

**All BNLN Corpus list:**  
<https://github.com/sagorbrur/bnlp/tree/master/bnlp/corpus>

**Pre-trained models-**

**Bengali SentencePiece:** <https://github.com/sagorbrur/bnlp/tree/master/model>

**Bengali POS Tagging:**  
[https://github.com/sagorbrur/bnlp/blob/master/model/bn\\_pos.pkl](https://github.com/sagorbrur/bnlp/blob/master/model/bn_pos.pkl)

**Bengali POS (Parts-Of-Speech) Tagging using Indian corpus-**  
<https://medium.com/analytics-vidhya/bengali-pos-part-of-speech-tagging-using-indian-corpus-e85f47d3ad65>

**Required Training sets:** <https://github.com/sagorbrur/bnlp/tree/master/model>



## **Towards POS Tagging Methods for Bengali Language: A Comparative Analysis.**

Department of Computer Science & Engineering, Chittagong University of Engineering & Technology, Chittagong-4349 Bangladesh. Authors:

- **Fatima Jahara**, [fatimajahara@ieee.org](mailto:fatimajahara@ieee.org)
- **Adrita Barua**, [adrita766@gmail.com](mailto:adrita766@gmail.com)
- **MD. Asif Iqbal**, [asifiqbalsagor123@gmail.com](mailto:asifiqbalsagor123@gmail.com)
- **Avishek Das**, [avishek.das.ayan@gmail.com](mailto:avishek.das.ayan@gmail.com)
- **Omar Sharif**, [omar.sharif@cuet.ac.bd](mailto:omar.sharif@cuet.ac.bd)
- **Mohammed Moshiul Hoque**, [moshiul\\_240@cuet.ac.bd](mailto:moshiul_240@cuet.ac.bd)
- **Iqbal H. Sarker**, [iqbal@cuet.ac.bd](mailto:iqbal@cuet.ac.bd)

**Part-of-Speech Tagging for Bengali.** Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur. Authors:

- **Sandipan Dandapat**,
- **Prof. Sudeshna Sarkar**,
- **Prof. Anupam Basu**,