
bnlp Documentation

Release latest

Apr 29, 2021

Contents

1	Installation	3
2	Pretrained Model	5
2.1	Download Link	5
2.2	Training Details	5
3	Tokenization	7
4	Word Embedding	9
5	Bengali POS Tagging	13
6	Bengali NER	15
7	Bengali Corpus Class	17
8	Contributor Guide	19

BNLP is a natural language processing toolkit for Bengali Language. This tool will help you to **tokenize Bengali text**, **Embedding Bengali words**, **Bengali POS Tagging**, **Construct Neural Model** for Bengali NLP purposes.

CHAPTER 1

Installation

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

```
pip install bnlp_toolkit
```

or Upgrade

```
pip install -U bnlp_toolkit
```


2.1 Download Link

- Bengali SentencePiece
- Bengali Word2Vec
- Bengali FastText
- Bengali GloVe Wordvectors
- Bengali POS Tag model
- Bengali NER model

2.2 Training Details

- Sentencepiece, Word2Vec, Fasttext, GloVe model trained with **Bengali Wikipedia Dump Dataset**
 - Bengali Wiki Dump
- SentencePiece Training Vocab Size=50000
- Fasttext trained with total words = 20M, vocab size = 1171011, epoch=50, embedding dimension = 300 and the training loss = 0.318668,
- Word2Vec word embedding dimension = 100, min_count=5, window=5, epochs=10
- To Know Bengali GloVe Wordvector and training process follow [this](#) repository
- Bengali CRF POS Tagging was training with [nltr](#) dataset with 80% accuracy.
- Bengali CRF NER Tagging was train with [this](#) data with 90% accuracy.

- **Basic Tokenizer**

```
from bnlp import BasicTokenizer
basic_t = BasicTokenizer()
raw_text = " "
tokens = basic_t.tokenize(raw_text)
print(tokens)

# output: ["", "", "", "", ""]
```

- **NLTK Tokenization**

```
from bnlp import NLTKTokenizer

bnltk = NLTKTokenizer()

text = " ?"

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

# output
# word_token: ["", "", "", "", "", "", "", "", "", "", "", "", "", "", "??"]
# sentence_token: [" ", " ", " ", " " " ?"]
```

- **Bengali SentencePiece Tokenization**

- tokenization using trained model

```
from bnlp import SentencepieceTokenizer

bsp = SentencepieceTokenizer()
model_path = "./model/bn_spm.model"
```

(continues on next page)

(continued from previous page)

```
input_text = "      "  
tokens = bsp.tokenize(model_path, input_text)  
print(tokens)
```

– Training SentencePiece

```
from bnlp import SentencepieceTokenizer  
  
bsp = SentencepieceTokenizer()  
data = "sample.txt"  
model_prefix = "test"  
vocab_size = 5  
bsp.train(data, model_prefix, vocab_size)
```

- **Bengali Word2Vec**

- Generate Vector using pretrain model

```
from bnlp import BengaliWord2Vec

bvw = BengaliWord2Vec()
model_path = "model/bengali_word2vec.model"
word = ''
vector = bwv.generate_word_vector(model_path, word)
print(vector.shape)
print(vector)
```

- Find Most Similar Word Using Pretrained Model

```
from bnlp import BengaliWord2Vec

bvw = BengaliWord2Vec()
model_path = "model/bengali_word2vec.model"
word = ''
similar = bwv.most_similar(model_path, word, topn=10)
print(similar)
```

- Train Bengali Word2Vec with your own data Train Bengali word2vec with your custom raw data or tokenized sentences. custom tokenized sentence format example: sentences = [['', '', '', ''], ['', '', '', '']]

Check [gensim word2vec api](#) for details of training parameter

```
from bnlp import BengaliWord2Vec

bvw = BengaliWord2Vec()
data_file = "test.txt"
model_name = "test_model.model"
vector_name = "test_vector.vector"
bvw.train(data_file, model_name, vector_name)
```

- Pre-train or resume word2vec training with same or new corpus or tokenized sentences

Check [gensim word2vec api](#) for details of training parameter

```
from bnlp import BengaliWord2Vec
bvw = BengaliWord2Vec()

trained_model_path = "mytrained_model.model"
data_file = "raw_text.txt"
model_name = "test_model.model"
vector_name = "test_vector.vector"
bvw.pretrain(trained_model_path, data_file, model_name, vector_name, epochs=5)
```

- **Bengali FastText** Install fasttext first by pip install fasttext

- Generate Vector Using Pretrained Model

```
from bnlp.embedding.fasttext import BengaliFasttext

bft = BengaliFasttext()
word = ""
model_path = "model/bengali_fasttext.bin"
word_vector = bft.generate_word_vector(model_path, word)
print(word_vector.shape)
print(word_vector)
```

- Train Bengali FastText Model

Check [fasttext documentation](#) for details of training parameter

```
from bnlp.embedding.fasttext import BengaliFasttext

bft = BengaliFasttext()
data = "data.txt"
model_name = "saved_model_wiki.bin"
epoch = 10
bft.train(data, model_name, epoch)
```

- Generate Vector File from Fasttext Binary Model

```
from bnlp.embedding.fasttext import BengaliFasttext

bft = BengaliFasttext()

model_path = "mymodel.bin"
out_vector_name = "myvector.txt"
bft.bin2vec(model_path, out_vector_name)
```

- **Bengali GloVe Word Vectors**

We trained glove model with bengali data(wiki+news articles) and published bengali glove word vectors</br> You can download and use it on your different machine learning purposes.

```
from bnlp import BengaliGlove

bng = BengaliGlove()
glove_path = "bn_glove.39M.100d.txt"
word = ""
res = bng.closest_word(glove_path, word)
```

(continues on next page)

(continued from previous page)

```
print(res)
vec = bng.word2vec(glove_path, word)
print(vec)
```

Bengali POS Tagging

- Bengali CRF POS Tagging
- Find Pos Tag Using Pretrained Model

```
from bnlp import POS
bn_pos = POS()
model_path = "model/bn_pos_model.pkl"
text = " "
res = bn_pos.tag(model_path, text)
print(res)
# [(' ', 'PPR'), (' ', 'NC'), (' ', 'VM'), (' ', 'PU')]
```

- Train POS Tag Model

```
from bnlp import POS
bn_pos = POS()
model_name = "pos_model.pkl"
tagged_sentences = [[(' ', 'JJ'), (' ', 'NC'), ('-', 'PU'), (' ', 'JJ'), (' ', 'CCD'),
→ (' ', 'JJ'), (' ', 'NC'), (' ', 'PU'), (' ', 'NC'), (' ', 'PU'), (' ', 'NC'), (' ',
→ 'CCD'), (' ', 'NC'), (' ', 'CCD'), (' ', 'NC'), (' ', 'PU')], [(' ', 'NC'), (' ', 'PP
→ '), (' ', 'JQ'), (' ', 'JQ'), (' ', 'JQ'), (' ', 'CCL'), (' ', 'JJ'), (' ', 'VM'), ('
→ ', 'PU')]]

bn_pos.train(model_name, tagged_sentences)
```


- Bengali CRF NER
- Find NER Tag Using Pretrained Model

```
from bnlp import ner
bn_ner = NER()
model_path = "model/bn_pos_model.pkl"
text = " "
res = bn_ner.tag(model_path, text)
print(res)
# [(' ', 'O'), (' ', 'S-LOC'), (' ', 'O')]
```

- Train NER Model

```
from bnlp import NER
bn_ner = NER()
model_name = "ner_model.pkl"
tagged_sentences = [[(' ', 'O'), (' ', 'O'), (' ', 'O'), (' ', 'S-PER'), (' ', 'B-PER'), (' '
→', 'I-PER'), (' ', 'E-PER'), (' ', 'O'), (' ', 'O'), (' ', 'O'), (' ', 'O'), (' ', 'O')]]

bn_ner.train(model_name, tagged_sentences)
```

Bengali Corpus Class

- Stopwords and Punctuations

```
from bnlp.corpus import stopwords, punctuations, letters, digits

print(stopwords)
print(punctuations)
print(letters)
print(digits)
```

- Remove Stopwords from text

```
from bnlp.corpus import stopwords
from bnlp.corpus.util import remove_stopwords

raw_text = ' '
result = remove_stopwords(raw_text, stopwords)
print(result)
# ['', ' ', '']
```


CHAPTER 8

Contributor Guide

Check [CONTRIBUTING.md](#) page for details.