
bendeeep Documentation

Release latest

Nov 02, 2020

Contents

1	Installation	3
1.1	Dependency	3
2	Pretrained Model	5
3	API	7
3.1	Sentiment Analysis	7
3.2	Machine Translation	8
4	References	9

BENDeep is a pytorch based deep learning solution for Bengali NLP Task like bengali translation, bengali sentiment analysis and so on.


```
pip install bendeep
```

1.1 Dependency

- pytorch 1.5.0+

CHAPTER 2

Pretrained Model

- Sentiment Analysis
- Translation Model

3.1 Sentiment Analysis

3.1.1 Analyzing Sentiment

This sentiment analysis model is a RNN based GRU model trained with [socian sentiment dataset](#) with loss 0.073 in 150 epochs. Dataset size: 4000 sentences

```
from bendeep import sentiment
model_path = "senti_trained.pt"
vocab_path = "vocab.txt"
text = "    "

sentiment.analyze(model_path, vocab_path, text)
```

3.1.2 Training Sentiment Model

To train this model you need a csv file with one column `review` means text and another column `sentiment` with 0 or 1, where 1 for positive and 0 for negative sentiment.

Example:

	review	sentiment
0		1
1		0

```
from bendeep import sentiment
data_path = "sentiment_data.csv"
sentiment.train(data_path)
# you can also pass these parameter
# sentiment.train(data_path, batch_size = 64, epochs=100, model_name="trained.pt")
```

after successfully training it will complete training and save model as `trained.pt` also save vocab file as `vocab.txt`

3.2 Machine Translation

3.2.1 Translate Bengali to English

This model is a seq2seq attentional model trained with [this](#) dataset with loss 0.0.

```
from bendeeep import translation
from bendeeep.translation import EncoderRNN
from bendeeep.translation import AttnDecoderRNN

data_path = "data/translation/eng-ben.txt"
encoder = "models/translation/encoder.pt"
decoder = "models/translation/decoder.pt"
input_sentence = " "
translation.bn2en(data_path, encoder, decoder, input_sentence)
# outupt
# >
# = i feel cold .
```

3.2.2 Training Translation Model

To train translation model you need a dataset in `.txt` format with tab separate input and target sentences.

Example:

```
from bendeeep import translation
from bendeeep.translation import EncoderRNN
from bendeeep.translation import AttnDecoderRNN

data_path = "data/translation/eng-ben.txt"
translation.training(data_path, iteration=75000)
```

after successfully training it will complete training and save encoder and decoder model as `encoder.pt`, `decoder.pt`. Also display some random evaluation results.

CHAPTER 4

References

- [pytorch](#)
- [pytorch tutorial](#)
- [en-bn dataset](#)
- [socian sentiment dataset](#)