

---

# **bert-embedding Documentation**

**Gary Lai**

**Nov 13, 2019**



<b>1 BertEmbedding</b>	<b>3</b>
<b>2 Available pre-trained BERT models</b>	<b>5</b>
2.1 Usage . . . . .	5
<b>3 Indices and tables</b>	<b>7</b>
<b>Python Module Index</b>	<b>9</b>
<b>Index</b>	<b>11</b>



BERT, published by Google, is a new way to obtain pre-trained language model word representation. Many NLP tasks benefit from BERT to get the SOTA.

The goal of this project is to obtain the sentence and token embedding from BERT's pre-trained model. In this way, instead of building and doing fine-tuning for an end-to-end NLP model, you can build your model by just utilizing the sentence or token embedding.

This project is implemented with @MXNet. Special thanks to @gluon-nlp team.



## BertEmbedding

BERT embedding.

```
class bert_embedding.bert.BertEmbedding (ctx=cpu(0), dtype='float32',
                                         model='bert_12_768_12',
                                         dataset_name='book_corpus_wiki_en_uncased',
                                         params_path=None, max_seq_length=25,
                                         batch_size=256)
```

Bases: object

Encoding from BERT model.

#### Parameters

- **ctx** (*Context.*) – running BertEmbedding on which gpu device id.
- **dtype** (*str*) – data type to use for the model.
- **model** (*str, default bert\_12\_768\_12.*) – pre-trained BERT model
- **dataset\_name** (*str, default book\_corpus\_wiki\_en\_uncased.*) – pre-trained model dataset
- **params\_path** (*str, default None*) – path to a parameters file to load instead of the pretrained model.
- **max\_seq\_length** (*int, default 25*) – max length of each sequence
- **batch\_size** (*int, default 256*) – batch size

```
__init__ (ctx=cpu(0), dtype='float32', model='bert_12_768_12',
          dataset_name='book_corpus_wiki_en_uncased', params_path=None, max_seq_length=25,
          batch_size=256)
```

Encoding from BERT model.

#### Parameters

- **ctx** (*Context.*) – running BertEmbedding on which gpu device id.
- **dtype** (*str*) –

- **type to use for the model.** (*data*) –
- **model** (*str*, *default bert\_12\_768\_12.*) – pre-trained BERT model
- **dataset\_name** (*str*, *default book\_corpus\_wiki\_en\_uncased.*) – pre-trained model dataset
- **params\_path** (*str*, *default None*) – path to a parameters file to load instead of the pretrained model.
- **max\_seq\_length** (*int*, *default 25*) – max length of each sequence
- **batch\_size** (*int*, *default 256*) – batch size

**embedding** (*sentences*, *oov\_way='avg'*, *filter\_spec\_tokens=True*)

Get tokens, tokens embedding

### Parameters

- **sentences** (*List[str]*) – sentences for encoding.
- **oov\_way** (*str*, *default avg.*) – use **avg**, **sum** or **last** to get token embedding for those out of vocabulary words
- **filter\_spec\_tokens** (*bool*) – filter [CLS], [SEP] tokens.

**Returns** List of tokens, and tokens embedding

**Return type** List[(List[str], List[ndarray])]



---

## Available pre-trained BERT models

---

### 2.1 Usage

Example of using the large pre-trained BERT model from Google

```
from bert_embedding.bert import BertEmbedding

bert = BertEmbedding(model='bert_24_1024_16', dataset_name='book_corpus_wiki_en_cased
↪')
```

Source: gluonnlp



## CHAPTER 3

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



**b**

`bert_embedding.bert`, 3



## Symbols

`__init__()` (*bert\_embedding.bert.BertEmbedding*  
*method*), 3

## B

`bert_embedding.bert` (*module*), 3

`BertEmbedding` (*class in bert\_embedding.bert*), 3

## E

`embedding()` (*bert\_embedding.bert.BertEmbedding*  
*method*), 4