# NLP-tutorial

`nlp-tutorial` is a tutorial for who is studying NLP(Natural Language Processing) using **TensorFlow** and **Pytorch**. Most of the models in NLP were implemented with less than **100 lines** of code.(except comments or blank lines)

## Curriculum - (Example Purpose)

#### 1. Basic Embedding Model

- 1-1. NNLM(Neural Network Language Model) - **Predict Next Word**
  - Paper - A Neural Probabilistic Language Model(2003)
  - Colab - NNLM_Tensor.ipynb, NNLM_Torch.ipynb
- 1-2. Word2Vec(Skip-gram) - **Embedding Words and Show Graph**
  - Paper - Distributed Representations of Words and Phrases and their Compositionality(2013)
  - Colab - Word2Vec_Tensor(NCE_loss).ipynb, Word2Vec_Tensor(Softmax).ipynb, Word2Vec_Torch(Softmax).ipynb
- 1-3. FastText(Application Level) - **Sentence Classification**
  - Paper - Bag of Tricks for Efficient Text Classification(2016)
  - Colab - FastText.ipynb

#### 2. CNN(Convolutional Neural Network)

- 2-1. TextCNN - **Binary Sentiment Classification**
  - Paper - Convolutional Neural Networks for Sentence Classification(2014)
  - Colab - TextCNN_Tensor.ipynb, TextCNN_Torch.ipynb
- 2-2. DCNN(Dynamic Convolutional Neural Network)

#### 3. RNN(Recurrent Neural Network)

- 3-1. TextRNN - **Predict Next Step**
  - Paper - Finding Structure in Time(1990)
  - Colab - TextRNN_Tensor.ipynb, TextRNN_Torch.ipynb

- 3-2. TextLSTM - **Autocomplete**
  - Paper - LONG SHORT-TERM MEMORY(1997)
  - Colab - TextLSTM_Tensor.ipynb, TextLSTM_Torch.ipynb
- 3-3. Bi-LSTM - **Predict Next Word in Long Sentence**
  - Colab - Bi_LSTM_Tensor.ipynb, Bi_LSTM_Torch.ipynb

#### 4. Attention Mechanism

- 4-1. Seq2Seq - **Change Word**
  - Paper - Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation(2014)
  - Colab - Seq2Seq_Tensor.ipynb, Seq2Seq_Torch.ipynb
- 4-2. Seq2Seq with Attention - **Translate**
  - Paper - Neural Machine Translation by Jointly Learning to Align and Translate(2014)
  - Colab - Seq2Seq(Attention)_Tensor.ipynb, Seq2Seq(Attention)_Torch.ipynb
- 4-3. Bi-LSTM with Attention - **Binary Sentiment Classification**
  - Colab - Bi_LSTM(Attention)_Tensor.ipynb, Bi_LSTM(Attention)_Torch.ipynb

#### 5. Model based on Transformer

- 5-1. The Transformer - **Translate**
  - Paper - Attention Is All You Need(2017)
  - Colab - Transformer_Torch.ipynb, Transformer(Greedy_decoder)_Torch.ipynb
- 5-2. BERT - **Classification Next Sentence & Predict Masked Tokens**
  - Paper - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding(2018)
  - Colab - BERT_Torch.ipynb

| Model | Example | Framework | Lines(torch/tensor) |
|---|---|---|---|
| NNLM | Predict Next Word | Torch, Tensor | 67/83 |
| Word2Vec(Softmax) | Embedding Words and Show Graph | Torch, Tensor | 77/94 |
| TextCNN | Sentence Classification | Torch, Tensor | 94/99 |
| TextRNN | Predict Next Step | Torch, Tensor | 70/88 |

| Model | Example | Framework | Lines(torch/tensor) |
|---|---|---|---|
| TextLSTM | Autocomplete | Torch, Tensor | 73/78 |
| Bi-LSTM | Predict Next Word in Long Sentence | Torch, Tensor | 73/78 |
| Seq2Seq | Change Word | Torch, Tensor | 93/111 |
| Seq2Seq with Attention | Translate | Torch, Tensor | 108/118 |
| Bi-LSTM with Attention | Binary Sentiment Classification | Torch, Tensor | 92/104 |
| Transformer | Translate | Torch | 222/0 |
| Greedy Decoder Transformer | Translate | Torch | 246/0 |
| BERT | how to train | Torch | 242/0 |

## Dependencies

- Python 3.5+
- Tensorflow 1.12.0+
- Pytorch 0.4.1+
- Plan to add Keras Version

## Author

- Tae Hwan Jung(Jeff Jung) @graykode
- Author Email : nlkey2022@gmail.com

- Acknowledgements to mojitok as NLP Research Internship.