

# Natural Language Processing with PyTorch

## PyTorch를 활용한 자연어처리

이 책은 딥러닝을 사용하여 자연어처리(Natural Language Processing)를 실무에 적용하고자 하는 입문자들을 대상으로 쓰여진 책입니다. 실무에 바로 투입 가능한 정도의 PyTorch 실습 코드를 활용하여 독자들의 이해를 돕고자 하였습니다. 시중에 널리 읽히고 있는 책들과 달리 딥러닝 자체에 대한 설명에 집중하기 보단, 자연어처리(특히 자연어생성, Natural Language Generation)에 더 집중하였습니다. 사실 인터넷이나 시중에는 자연어처리를 특정하여 다루는 자료는 다른 인공지능 분야(컴퓨터 비전 등)에 비하여 적기 때문에, 자연어처리를 공부하고자 하는 이들은 어려움을 겪을 수 밖에 없었습니다. 따라서, 딥러닝을 활용하여 자연어처리 기초부터 고급 이론 및 실습까지 제공하고자 합니다.

실제 시스템을 구현하며 얻은 경험과 인사이트들을 공유하고자 하였고, 배경이 되는 수학적 이론에서부터, 실제 PyTorch 코드, 그리고 실전에서의 꼭 필요한 개념들을 담을 수 있도록 하였습니다. 따라서, 현재 딥러닝을 활용한 State of the Art 기술 뿐만 아니라, 딥러닝 이전의 기존의 전통적인 방식부터 차근차근 설명하여, 왜 이 기술이 필요하고, 어떻게 발전 해 왔으며, 어떤 부분이 성능 개선을 만들어냈는지 쉽게 이해할 수 있도록 설명하고자 합니다.

- Github Repo: [https://github.com/kh-kim/nlp\\_with\\_pytorch](https://github.com/kh-kim/nlp_with_pytorch)
- Gitbook: <https://kh-kim.gitbook.io/natural-language-processing-with-pytorch/>

## 현재 작성 중인 챕터

대부분의 챕터는 작성이 완료되었으며, 아래의 챕터가 아직 작성 중입니다.

- Word Embedding Vector

## Pre-requisites

- Python
- Calculus, Linear Algebra
- Probability and Statistics
- Basic Machine Learning

- Objective / loss function
- Linear / logistic regression
- Gradient descent
- Basic Deep Learning
- Back-propagation
- Activation function

지은이: 김기현(Kim, Ki Hyun)



Figure 1:

## 연락처

Name	Kim, Ki Hyun
email	pointzz.ki@gmail.com
linkedin	<a href="https://www.linkedin.com/in/ki-hyun-kim/">https://www.linkedin.com/in/ki-hyun-kim/</a>
github:	<a href="https://github.com/kh-kim/">https://github.com/kh-kim/</a>

## 이력

- Principal Machine Learning Engineer @ Makinarocks
- Machine Learning Researcher @ SKPlanet
- Neural Machine Translation: Global 11번가
- Machine Learning Engineer @ Ticketmonster
- Recommender System: TMON
- Researcher @ ETRI

- Automatic Speech Translation: GenieTalk [Android], [iOS], [TV Ads]

## 패스트캠퍼스 강의 소개



현재 이 책을 바탕으로 패스트캠퍼스에서 자연어처리 입문 캠프, 자연어처리 심화 캠프도 진행하고 있습니다.

이 저작물은 크리에이티브 커먼즈 저작자표시-비영리-동일조건변경허락(BY-NC-SA)에 따라 이용할 수 있습니다.



- 소개글
- Natural Language Processing with Deeplearning
- Intro
- Deeplearning
- Why NLP is difficult
- Why Korean NLP is Hell
- Recent Trends
- Hello PyTorch
- Intro
- How to install
- PyTorch tutorial

- Word Senses: Similarity and Ambiguity
- Intro
- WordNet
- Appendix: TF-IDF
- How to get Similarity
- Word Sense Disambiguation
- Appendix: Monty-Hall Problem
- Selectional Preference
- Conclusion
- Preprocessing
- Intro
- Collecting corpus
- Cleaning corpus
- Tokenization (POS Tagging)
- Aligning parallel corpus
- Subword Segmentation
- Detokenization
- TorchText
- Word Embedding Vector
- Intro
- One-hot encoding
- Previous Methods
- Word2Vec
- GloVe
- FastText
- Doc2Vec
- Sequence Modeling
- Intro
- Recurrent Neural Network
- Long Short Term Memory
- Gated Recurrent Unit
- Gradient Clipping
- Text Classification
- Intro
- Naive Bayes
- Using RNN
- Using CNN
- Unsupervised Text Classification

- Language Modeling
- Intro
- n-gram
- Perplexity
- Appendix: Mean Square Error (MSE)
- n-gram Exercise
- Neural Network Language Model
- Applications
- Neural Machine Translation
- Intro
- Sequence-to-Sequence
- Attention
- Input Feeding
- Auto-regressive and Teacher Forcing
- Search
- Evaluation
- Source Code
- Advanced Topic on NMT
- Multilingual NMT
- Using Monolingual Corpora
- Fully Convolutional Seq2seq
- Transformer
- NLP with Reinforcement Learning
- Intro
- Math Basics
- Reinforcement Learning Basics
- Policy Gradients
- Reinforcement Learning on NLG
- Supervised NMT
- Unsupervised NMT
- Exploit Duality
- Duality
- Dual Supervised Learning
- Dual Unsupervised Learning
- Productization
- Pipeline
- Google's NMT
- Edinburgh's NMT

- Booking.com's NMT
- Microsoft's NMT
- References