Language and Computer

Fall 2023

100.130(001)

Course Information

- Instructor: Sangah Lee (Dept. of Linguistics, Seoul National University)
  (sanalee@snu.ac.kr)
- TA: Minji Kang (mnjkng@snu.ac.kr)
- Lecture: Mon, Wed 12:30–13:45 (bldg. 9, room 119)
- Textbooks: slides and supplementary materials provided https://web.stanford.edu/~jurafsky/slp3/
  + Jurafsky and Martin (2023 draft), "Speech and Language Processing"
- Make sure that you should be familiar with basic Python skills:
  At least you have to be able to use data structures, loops, and functions.
- Office Hour: Wed 14:00–16:00 (bldg. 3, room 311)
  Please make an appointment before visiting! https://calendly.com/sanalee/office-hours
- Language: Korean (English is allowed, if needed)

Objectives

- Introduction to fundamental notions and theories on CL and NLP
  - Focusing on data processing and deep learning models
- Development of programming and research abilities
  - understanding deep learning models
  - dealing with issues of CL and NLP
  - using Python-based tools (e.g. PyTorch)

Requirements

- Grade Policies: Relative Grading (A-F)

Item	Attendance	Assignment	Midterm	Final	Total
Rate (%)	10	30	30	30	100

- If you have a valid reason for absence, please submit:
  the relevant documents and the Attendance Acknowledgment Request Form
  (uploaded on the eTL page)
- Midterm: paper test
- Final: paper test
- Assignments: Python programming exercises
- If necessary, engineering majors can be evaluated separately as a group.

Syllabus

Week	Date	Topic
1	9/4, 9/6	Course Introduction, NLP Pipeline
2	9/11, 9/13	Basics of Text Processing, Encoding, csv, json
3	9/18, 9/20	Regular Expressions
4	9/25, 9/27	Text Tokenization, Numpy and Pandas
5	10/2, 10/4	Numpy, Pandas, PyTorch
6	10/11	Logistic Regression
7	10/16, 10/18	Logistic Regression
8	10/23, 10/25	Midterm Exam

Syllabus

Week	Date	Topic
9	10/30, 11/1	Deep Learning
10	11/6, 11/8	Feed Forward Neural Network
11	11/13, 11/15	Recurrent Neural Network
12	11/20, 11/22	Long Short Term Memory
13	11/27, 11/29	Convolutional Neural Network
14	12/4, 12/6	NLP Applications
15	12/11, 12/13	Final Exam

Environment Settings

- Make sure that you have environments for Python programming:
  - Google Colab https://colab.research.google.com/
  - Jupyter Notebook https://jupyter.org/

- Assignment codes will be basically based on .ipynb.
  - .py forms are also allowed: but the paths should be correct!

Computational Linguistics

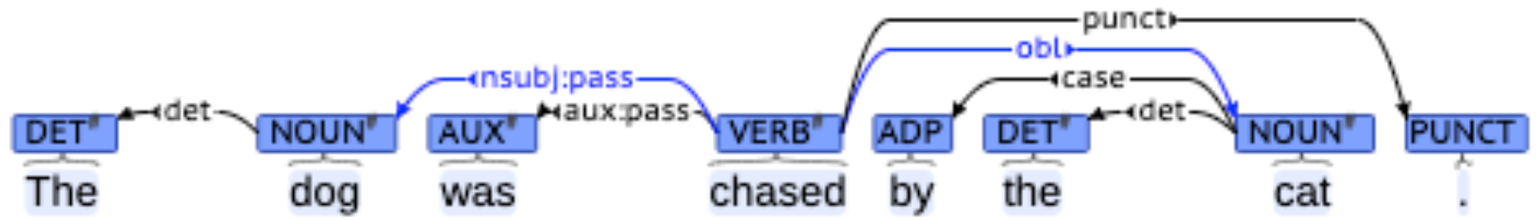
- A subfield of linguistics and computer science
  - concerned with the interaction of human language and computers
- Includes:
  - the analysis of written texts and spoken discourse
  - the translation of text and speech from one language into another
  - the use of human languages for communication between computers and people
  - the modeling and testing of linguistic theories
- Statistical analysis of written texts and spoken discourse
  - analysis on corpus: relative frequencies or collocation of letters, sounds, morphemes, words, …

Fromkin et al. (2018). "An Introduction to Language". Cengage Learning Australia.
https://medium.com/dair-ai/building-your-first-wordcloud-with-google-colaboratory-and-python-b3c8c1be5883

Computational Linguistics

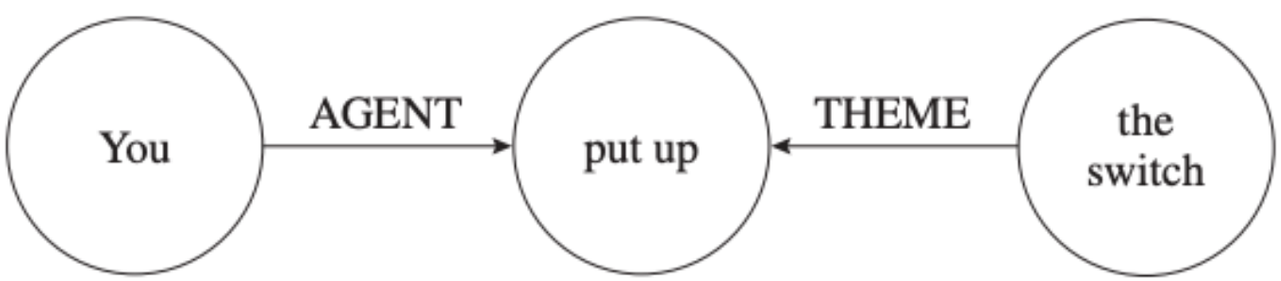
- The interaction between language and computers in all dimensions
  - Computational Phonetics and Phonology

    - Speech Recognition, Speech Synthesis (Text-to-Speech)
  - Computational Morphology: processing of word structures
  - Computational Syntax



https://universaldependencies.org/

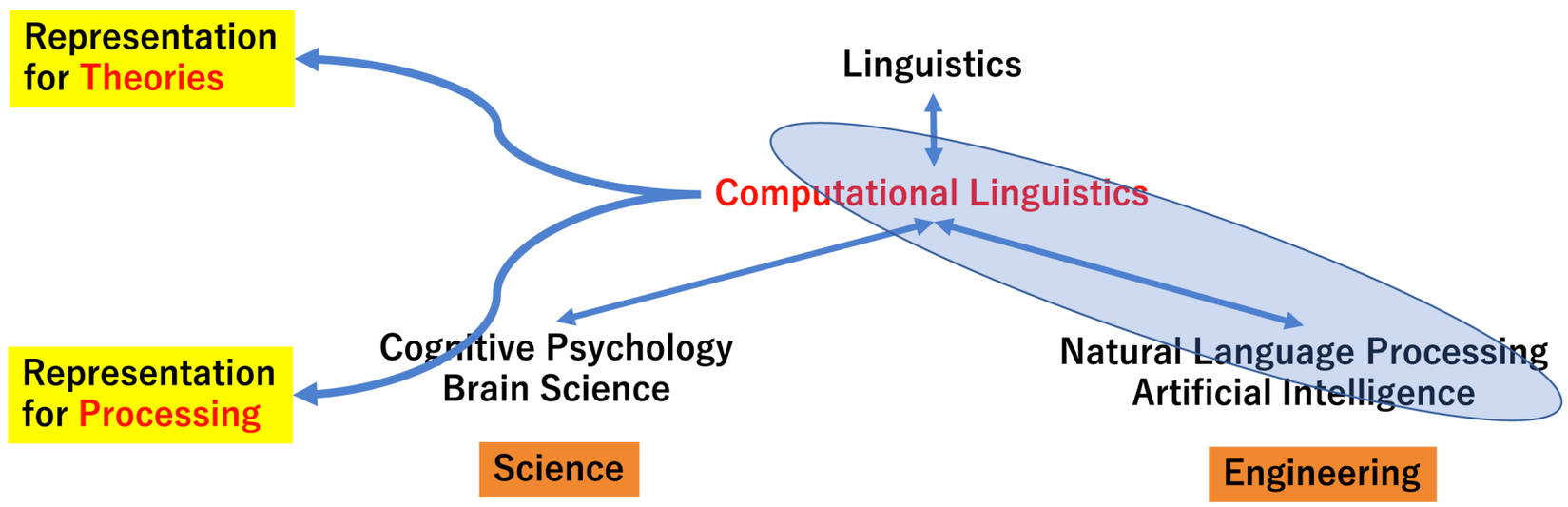
  - Computational Semantics: speech understanding and generation



  - Computational Pragmatics: sentence disambiguation, coreference resolution, ⋯

Natural Language Processing

- A schematic view of research disciplines: CL and NLP



how language is processed in our minds or our brains

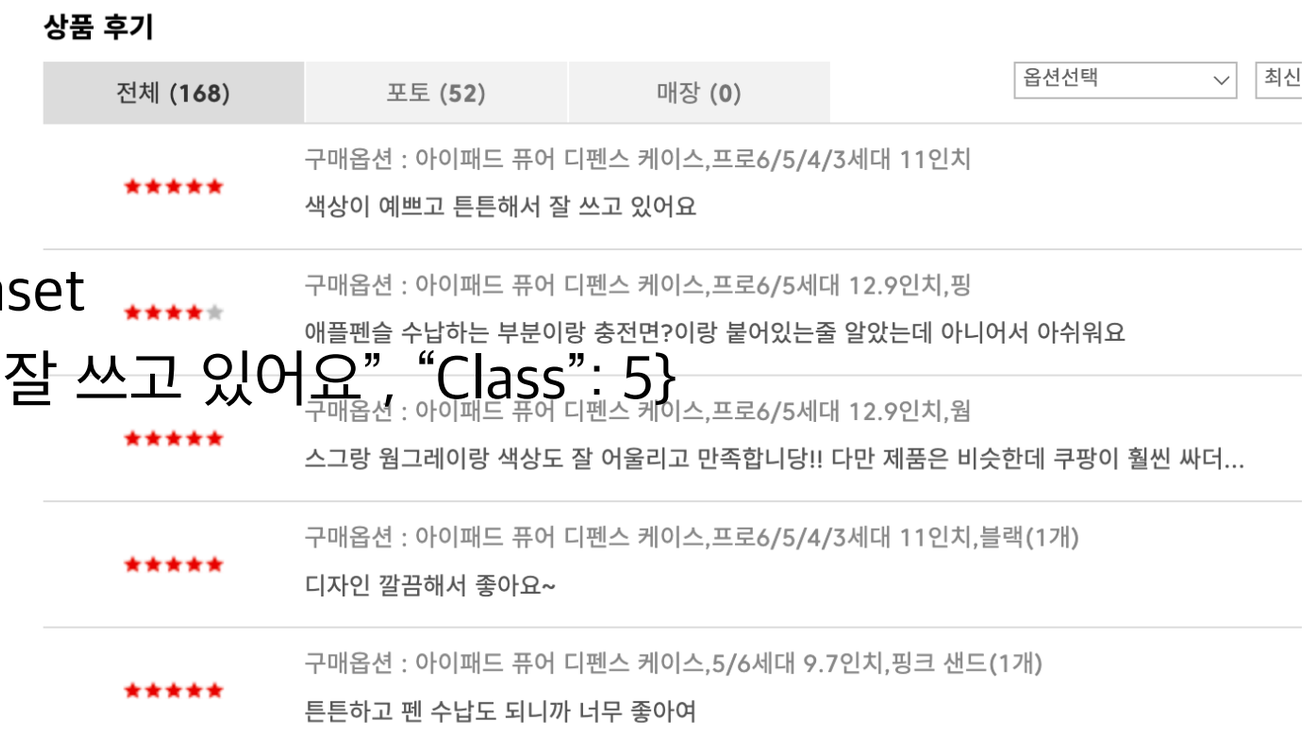
how computer systems should be designed to process language efficiently and effectively

- NLP may be included either in CL and other fields as their subfield.
  (Any other fields can deal with and utilize language data.)

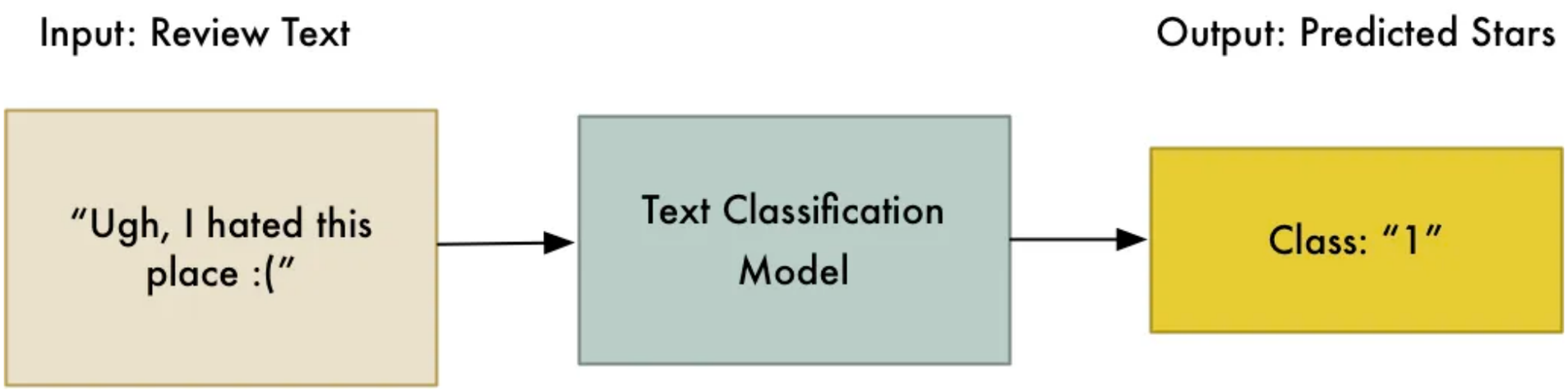
Tsujii (2021). "Natural Language Processing and Computational Linguistics". Computational Linguistics 2021. 47 (4): pp. 707-727.

Pipeline

- Text processing: raw texts —> dataset
  e.g. {“text”: “색상이 예쁘고 튼튼해서 잘 쓰고 있어요”, “Class”: 5}



- Model construction and training
- Inference

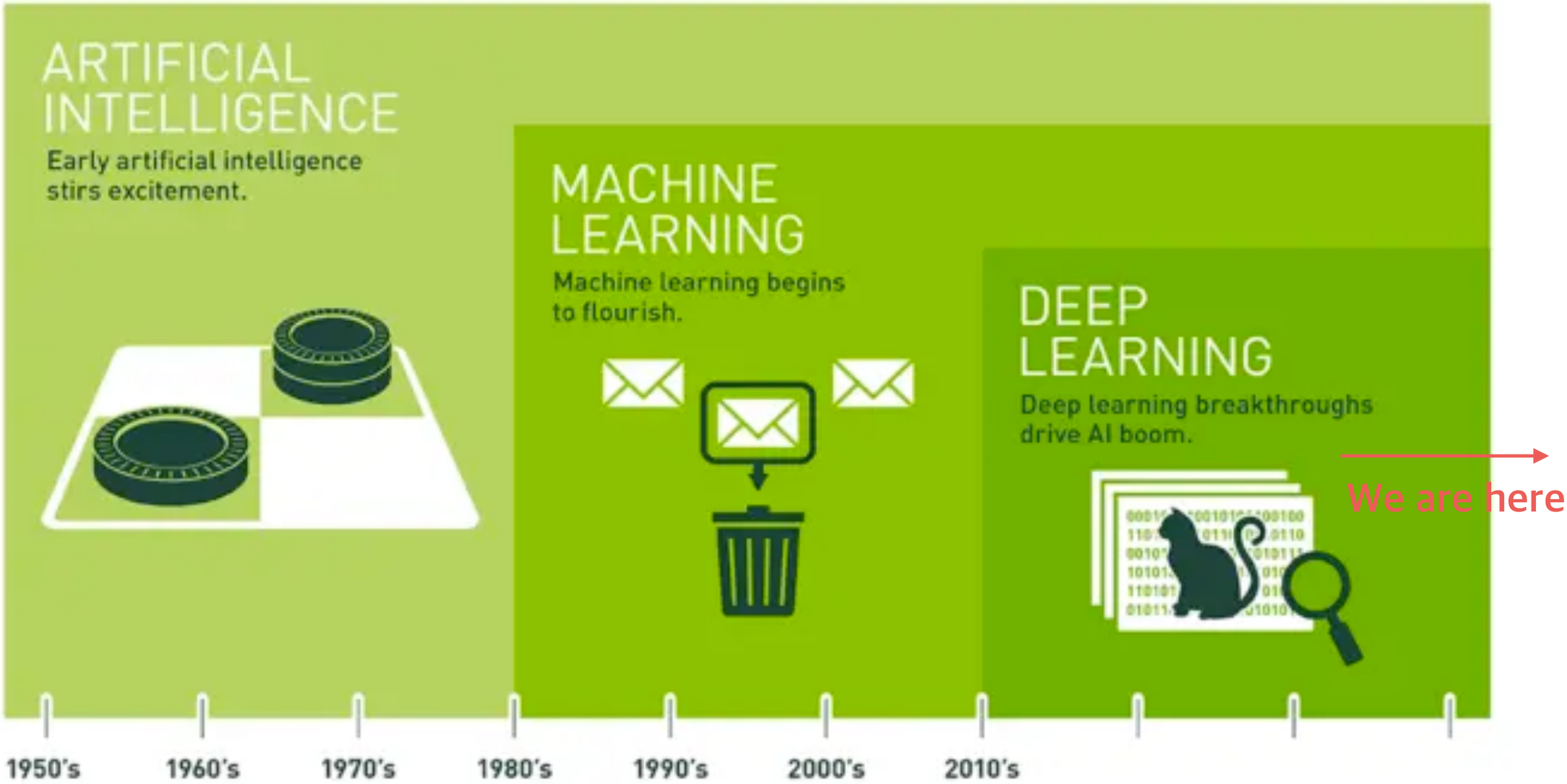


Text Processing

- Collecting and polishing texts to use for modeling
  - 데이터 크롤링(crawling), 직접 수집/구축, 첨가(augmentation) 등
  - 수집된 텍스트 내에서 필요 없는 내용 삭제
  - 개인정보 처리(masking)
  - 적절한 단위로 분절 (문장, 단어, 형태소, 문자, …)

- Obtaining a structured dataset
  - csv, json
  - Pandas

Deep Learning



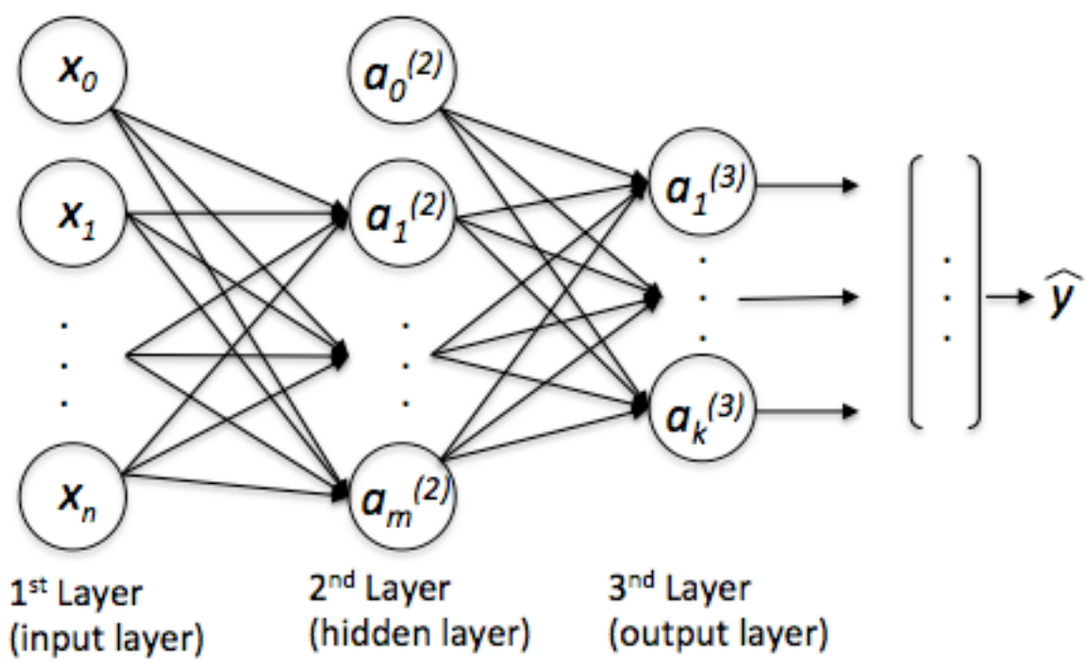
Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

Deep Learning

- A subset of Artificial Intelligence and Machine Learning
- Human neuron을 단순화하여 컴퓨팅/계산을 위한 단위로 사용하는 것
- Neuron들이 복잡하게 연결되어 있는 Neural Network
  - Neural Network들이 여러 층(layer)으로 연결되어 있음 -> Deep Learning
  - 여러 층위의 뉴런 사이의 계산을 통해 데이터의 feature들을 자동 연산함

Input: 나는 이 수업이 너무 좋아



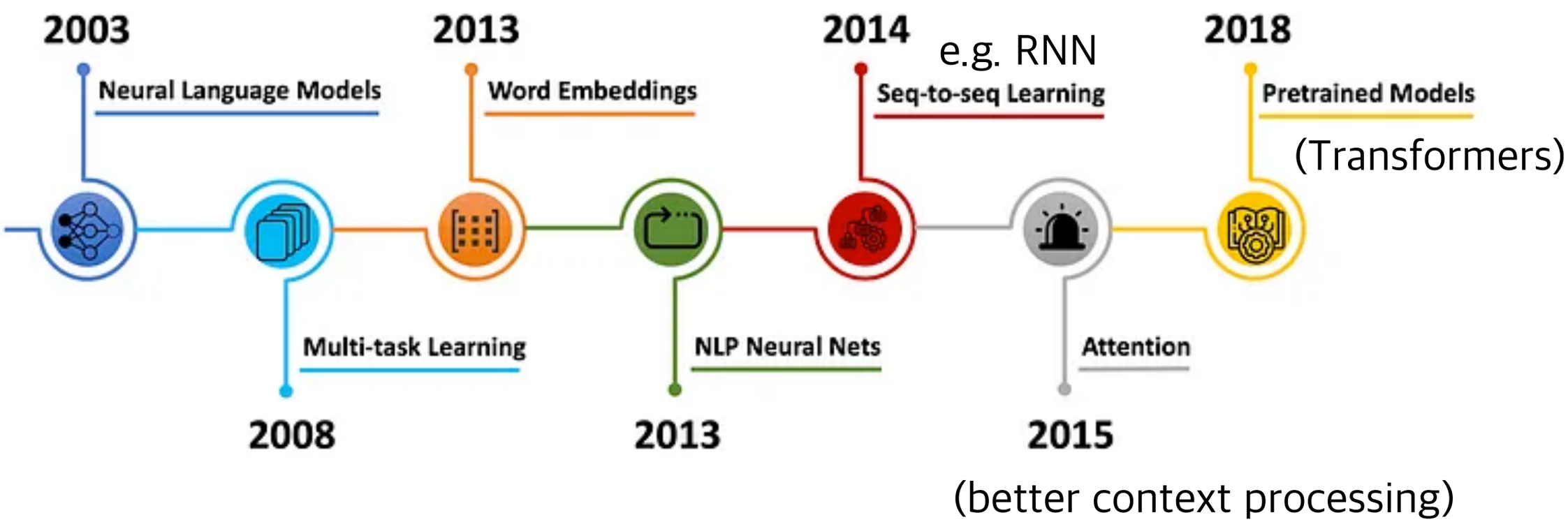
Schematic of a multi-layer perceptron.

Output:
P(진짜) vs. P(거짓말)

https://sebastianraschka.com/faq/docs/logisticregr-neuralnet.html

Deep Learning

- In NLP:
  - Neural models automatically learn low-dimensional continuous vectors from data as task-specific features.
  - capturing semantic meanings of words, phrases, and sentences, ⋯
  - CNN, RNN, LSTM, ⋯ models for various NLP tasks



Han et al. (2021). "Pre-Trained Models: Past, Present and Future". arXiv:2106.07319.
https://medium.com/@antoine.louis/a-brief-history-of-natural-language-processing-part-2-f5e575e8e37

PyTorch

PyTorch
https://pytorch.org/

- An open-source library for machine learning and deep learning applications
  - can use GPU accelerators -> faster!
  - 기본 자료 구조: Tensor (Numpy의 array와 유사, 호환)
  - 여러 기계학습 모델, 딥러닝 모델, 학습과 검증 등에 필요한 요소 등이 구현되어 있음

```
from torch.utils.data import DataLoader

train_dataloader = DataLoader(training_data, batch_size=64, shuffle=True)
test_dataloader = DataLoader(test_data, batch_size=64, shuffle=True)
```

데이터 로드, 구조화

```
import torch
from torch import nn
```

```
class NeuralNetwork(nn.Module):
    def __init__(self):
        super().__init__()
        self.flatten = nn.Flatten()
        self.linear_relu_stack = nn.Sequential(
            nn.Linear(28*28, 512),
            nn.ReLU(),
            nn.Linear(512, 512),
            nn.ReLU(),
            nn.Linear(512, 10),
        )

    def forward(self, x):
        x = self.flatten(x)
        logits = self.linear_relu_stack(x)
        return logits
```

class를 이용한 모델 정의

```
model = NeuralNetwork()
```

정의한 모델 객체 생성

```
logits = model(X)
```

모델에 input 넣고 계산