

Introduction to Natural Language Processing

Kim, Ki Hyun and June Oh

FastCampus

2018.09.08

Index

- Basics
 - Probability
 - Expectation and Sampling
 - Machine Learning
 - Information
 - Gradient Based Optimization
- Natural Language Processing
 - What is NLP?
 - History of Deep Learning
 - Why NLP is Hard?
 - Why Korean NLP is Hell?
 - Recent Trends of NLP

Random Variable

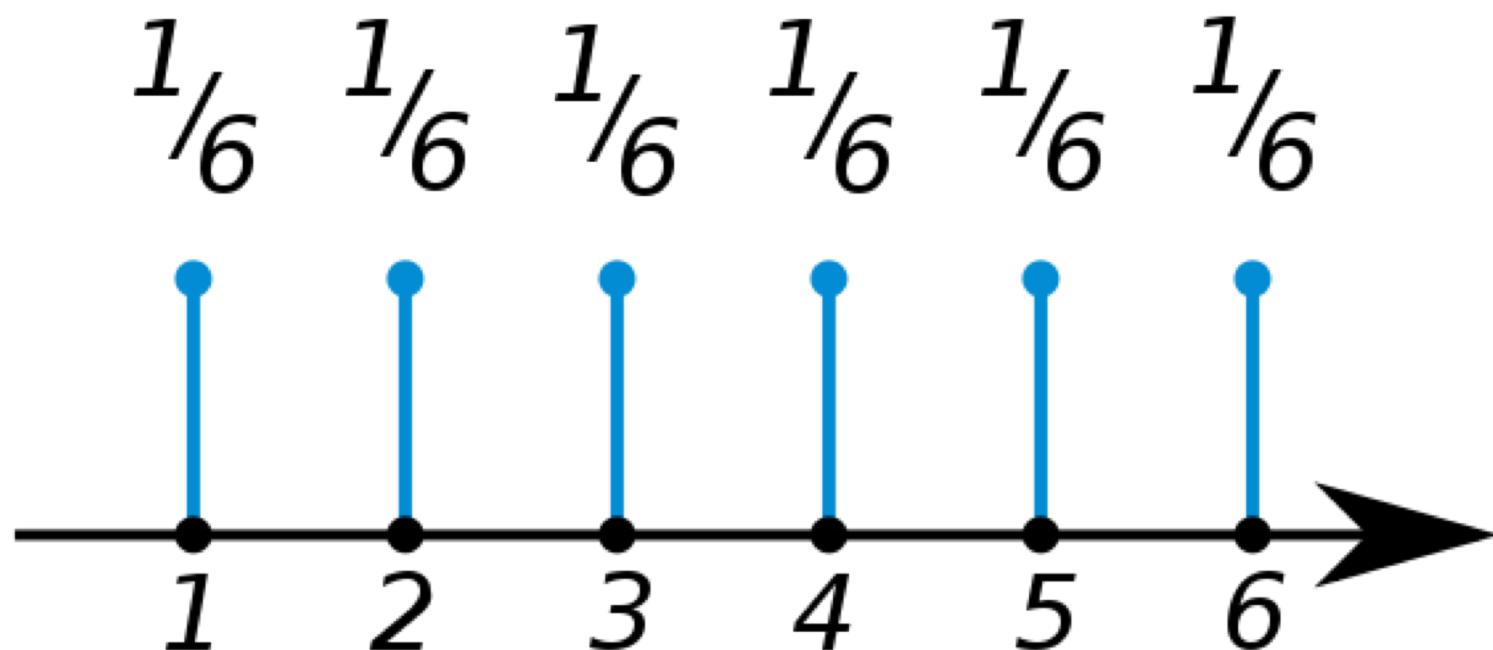
- Variable
 - It's like variable in programming, also.
 - It can contain values.
 - Discrete? Continuous?
- How to write in equation?

Discrete Variable

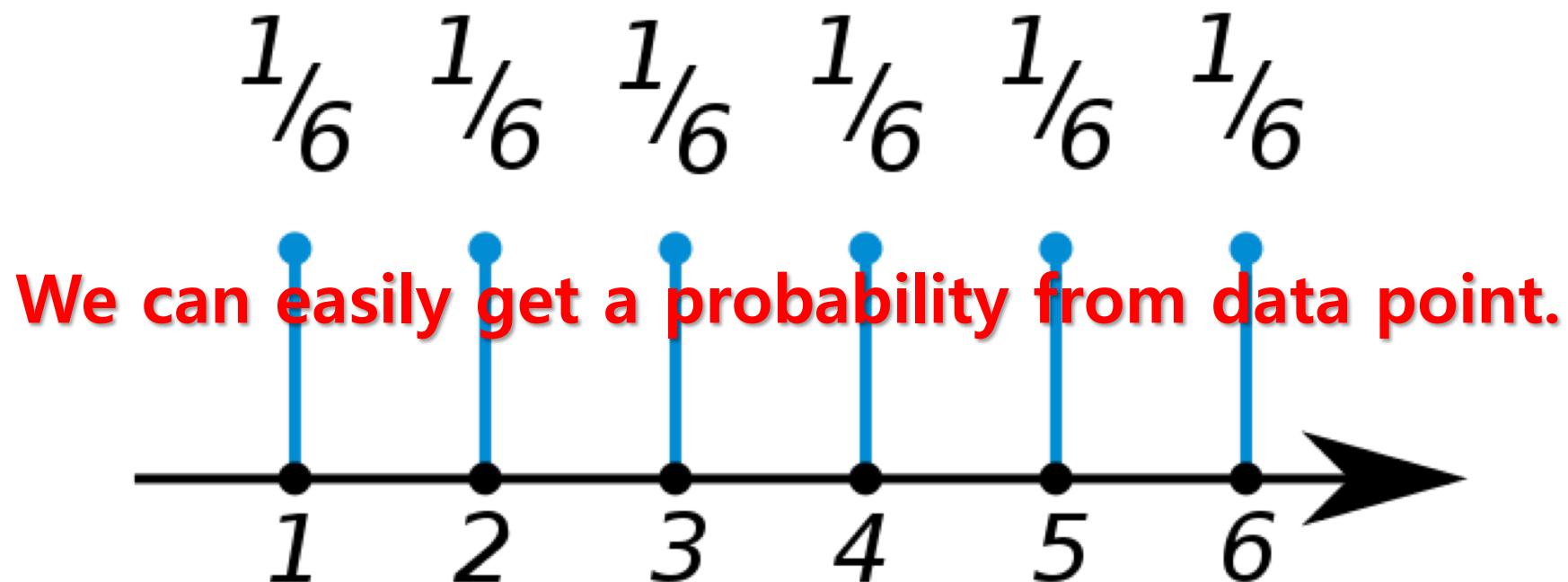
$$P(X = x)$$

$$\sum_{i=1}^N P(X = x_i) = \sum_{i=1}^N P(x_i) = 1$$

Probability Mass Function (PMF)



Probability Mass Function (PMF)



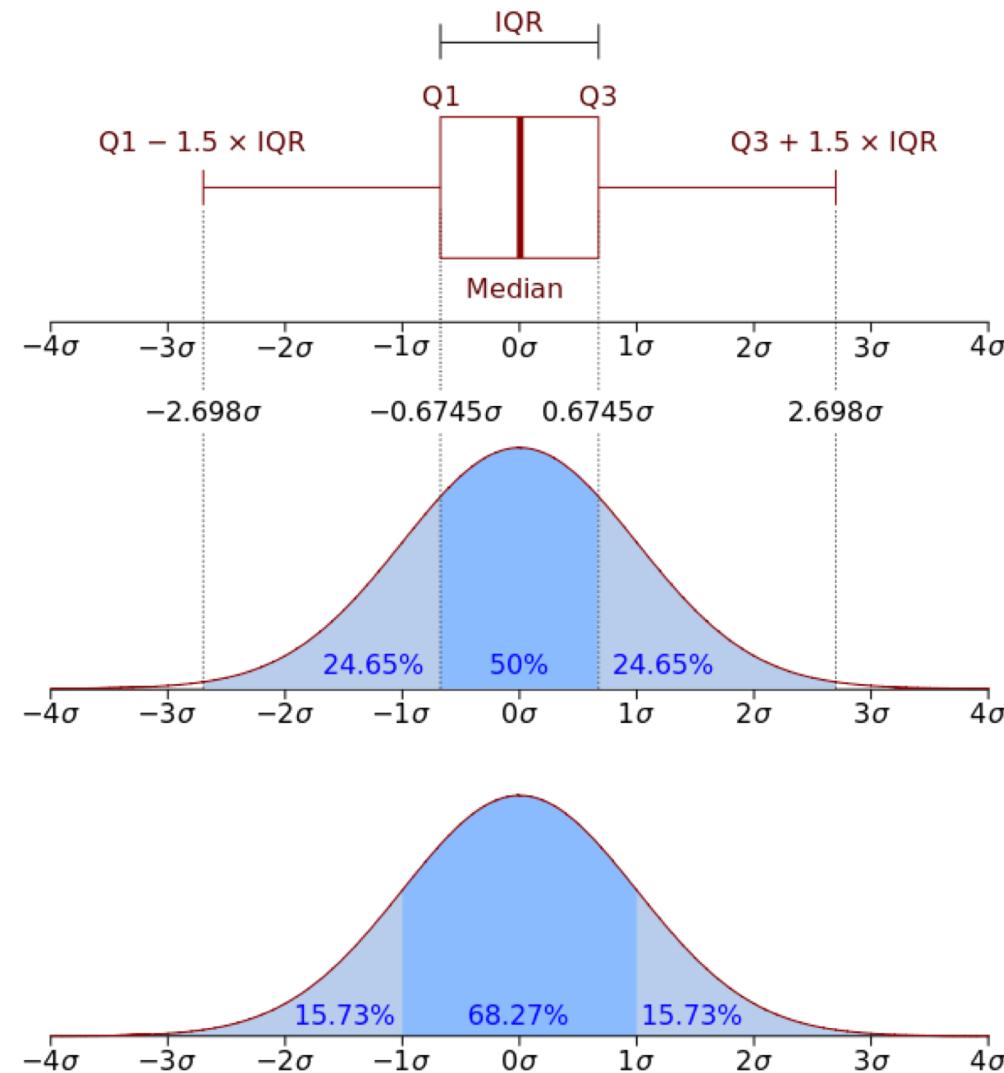
Continuous Variable

$$\forall x \in X, p(x) \geq 0.$$

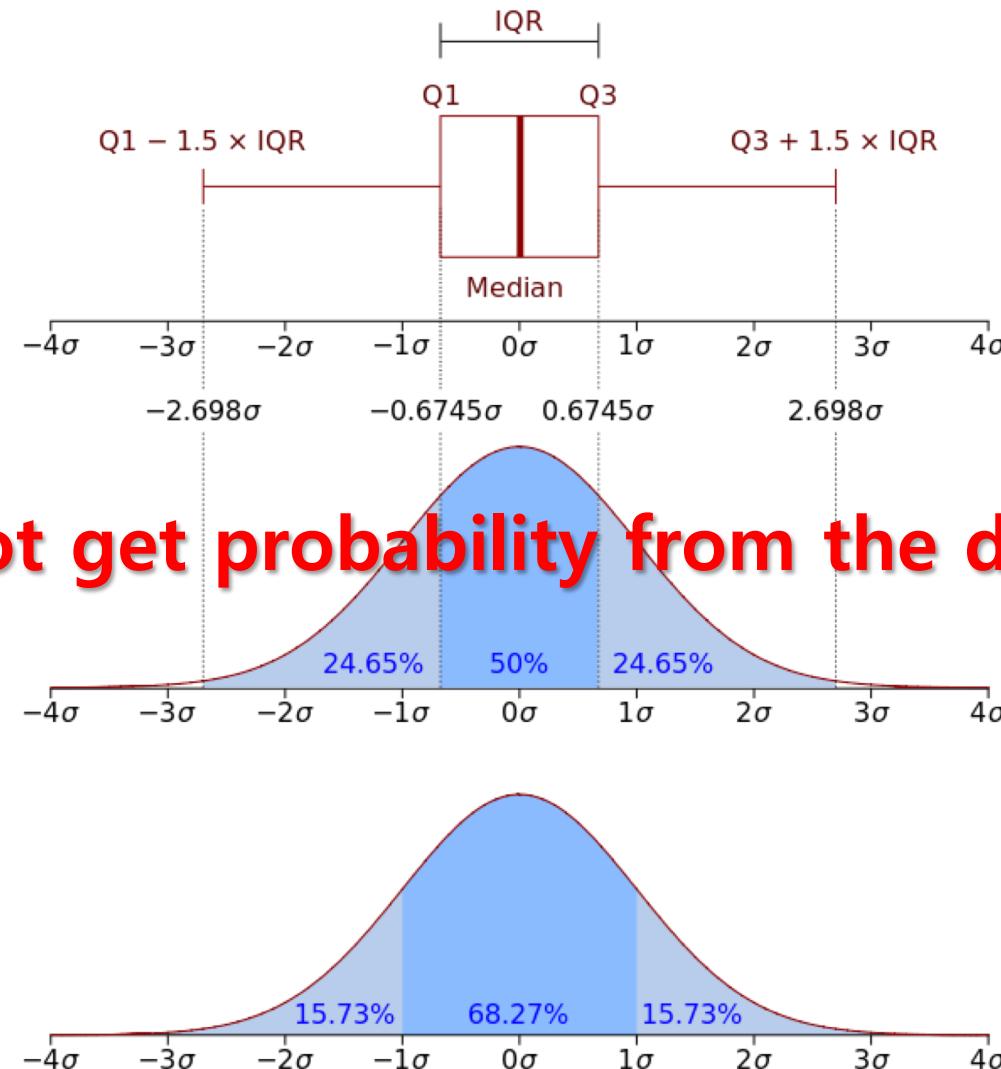
We do not require that $p(x) \leq 1$.

$$\int_{-\infty}^{\infty} p(x) dx = 1$$

Probability Density Function (PDF)



Probability Density Function (PDF)



Conditional Probability

$$P(A|B) = \frac{P(A, B)}{P(B)}$$

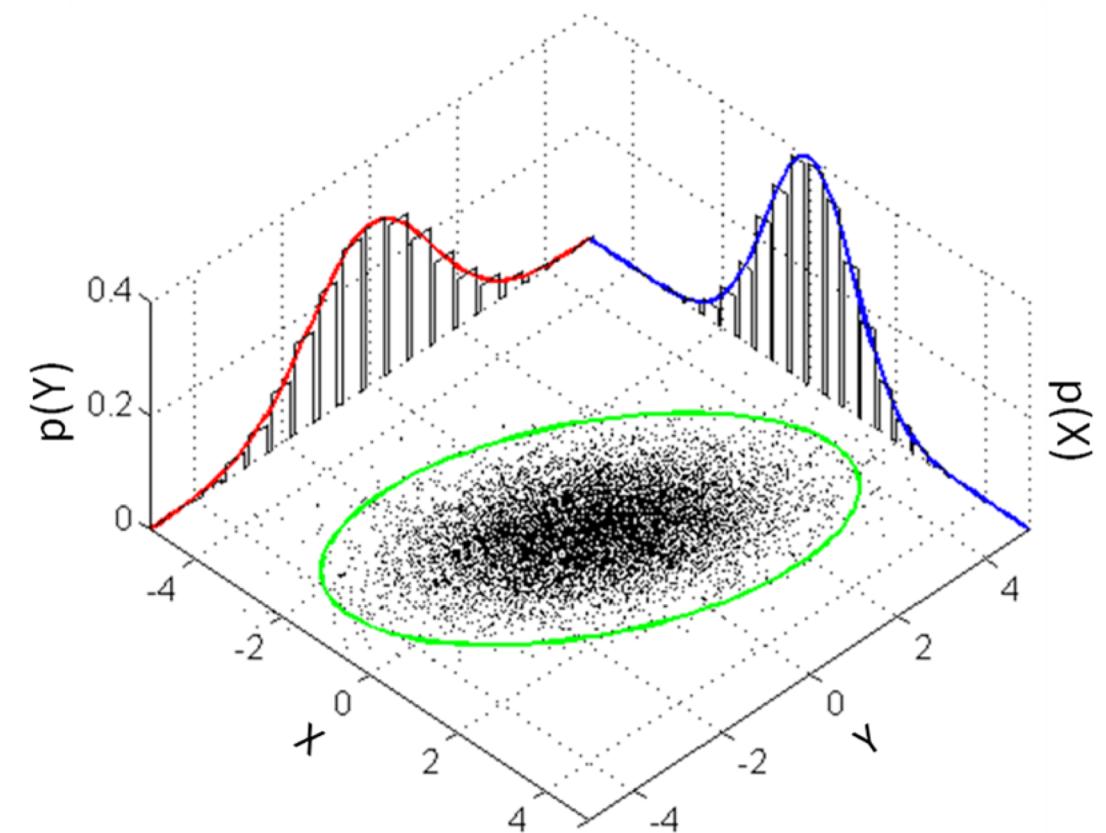
- Conditional Independence

$$P(A, B) = P(A)P(B)$$

$$P(A|B) = \frac{P(A, B)}{P(B)} = P(A)$$

Marginal Probability

$$P(y) = \sum_{x \in \mathcal{X}} P(x, y) = \sum_{x \in \mathcal{X}} P(y|x)P(x)$$

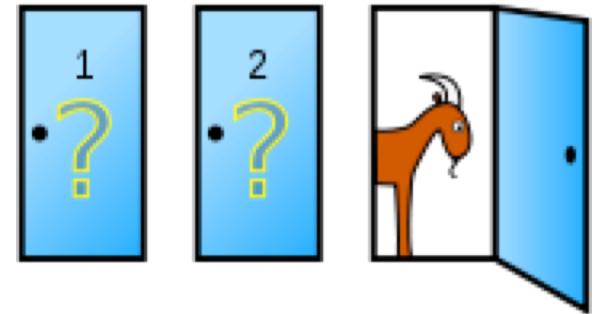


Bayes Theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Monty-hall Problem

- There are three doors, and you have to choose a door for prize. You can change your selection after MC shows that there is nothing behind the door, which is not selected by you. So, does it necessary to change your mind to get a prize?



- Random Variables
 - A: a door index what I selected at first time.
 - B: a door index what MC selected. MC will not open the answer.
 - C: a door index of the answer.

Monty-hall Problem

Monty-hall Problem

$$\begin{aligned} P(C = 0|A = 0, B = 1) &= \frac{P(A = 0, B = 1, C = 0)}{P(A = 0, B = 1)} \\ &= \frac{P(B = 1|A = 0, C = 0)P(A = 0, C = 0)}{P(A = 0, B = 1)} \\ &= \frac{P(B = 1|A = 0, C = 0)P(A = 0)P(C = 0)}{P(B = 1|A = 0)P(A = 0)} \\ &= \frac{\frac{1}{2} \times \frac{1}{3}}{\frac{1}{2}} = \frac{1}{3}, \end{aligned}$$

where $P(B = 1|A = 0, C = 0) = \frac{1}{2}$

$$P(B = 1, A = 0) = \frac{1}{2}, \quad P(C = 2) = \frac{1}{3}$$

Monty-hall Problem

$$\begin{aligned} P(C = 2|A = 0, B = 1) &= \frac{P(A = 0, B = 1, C = 2)}{P(A = 0, B = 1)} \\ &= \frac{P(B = 1|A = 0, C = 2)P(A = 0, C = 2)}{P(A = 0, B = 1)} \\ &= \frac{P(B = 1|A = 0, C = 2)P(A = 0)P(C = 2)}{P(B = 1|A = 0)P(A = 0)} \\ &= \frac{\frac{1}{2} \times \frac{1}{3}}{\frac{1}{2}} = \frac{2}{3}, \end{aligned}$$

where $P(B = 1, A = 0) = \frac{1}{2}$, $P(C = 2) = \frac{1}{3}$, and $P(B = 1|A = 0, C = 2) = 1$.

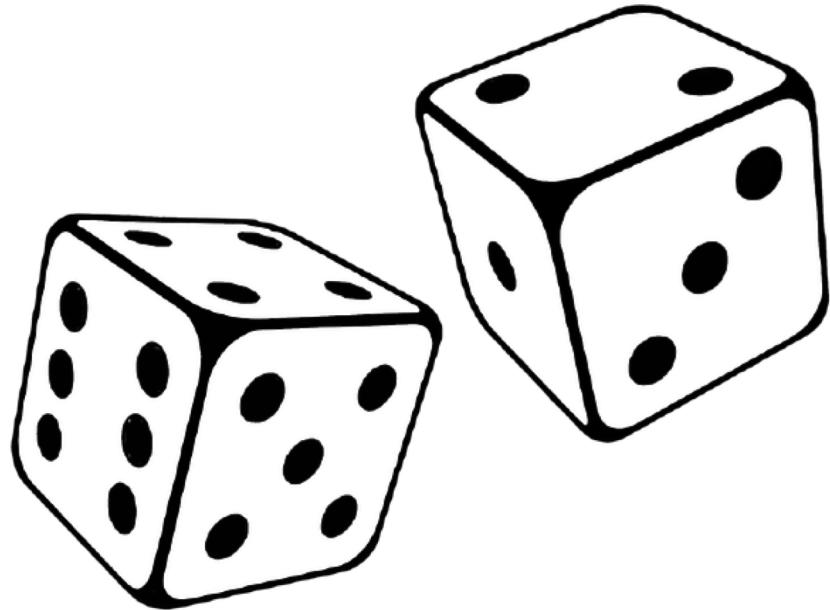
Index

- Basics
 - Probability
 - Expectation and Sampling
 - Machine Learning
 - Information
 - Gradient Based Optimization
- Natural Language Processing
 - What is NLP?
 - History of Deep Learning
 - Why NLP is Hard?
 - Why Korean NLP is Hell?
 - Recent Trends of NLP

Expectation

- 무엇을 선택하시겠습니까?
 - 1% 확률로 1억원
 - 0.001% 확률로 100억원
- 기대값 = 확률 * 보상

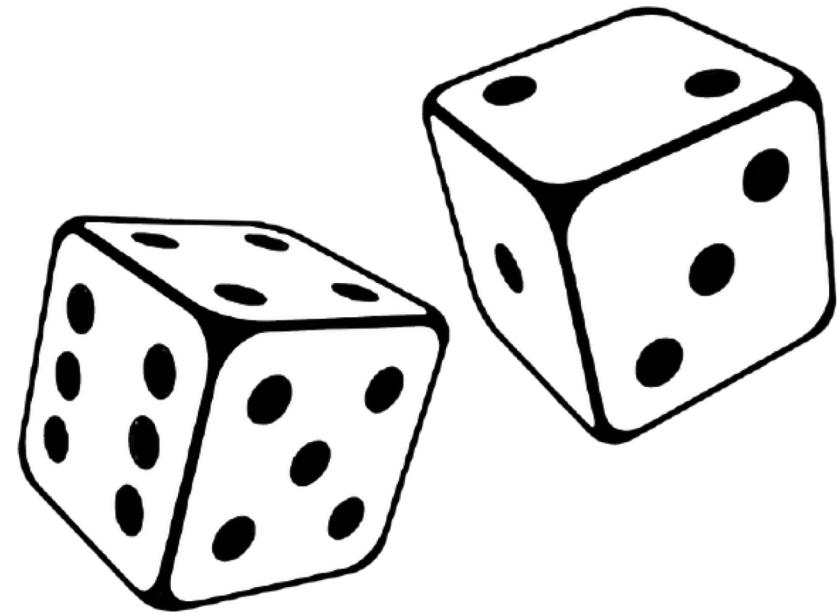
Expectation



expected reward from dice = $\sum_{x=1}^6 P(X = x) \times \text{reward}(x)$

where $P(x) = \frac{1}{6}, \forall x$ and $\text{reward}(x) = x$.

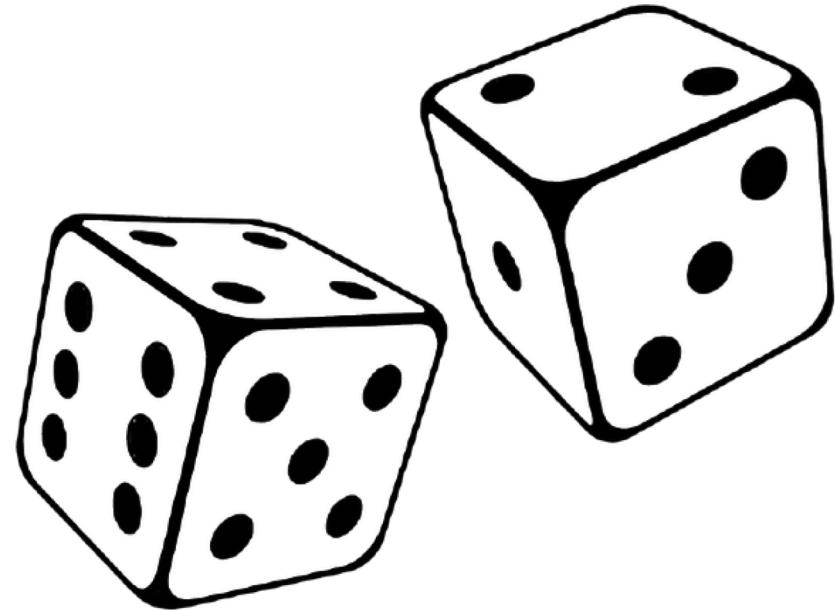
Expectation



$$\frac{1}{6} \times (1 + 2 + 3 + 4 + 5 + 6) = 3.5$$

$$E_{x \sim P(X)}[\text{reward}(x)] = \sum_{x=1}^6 P(X = x) \times \text{reward}(x) = 3.5$$

Expectation

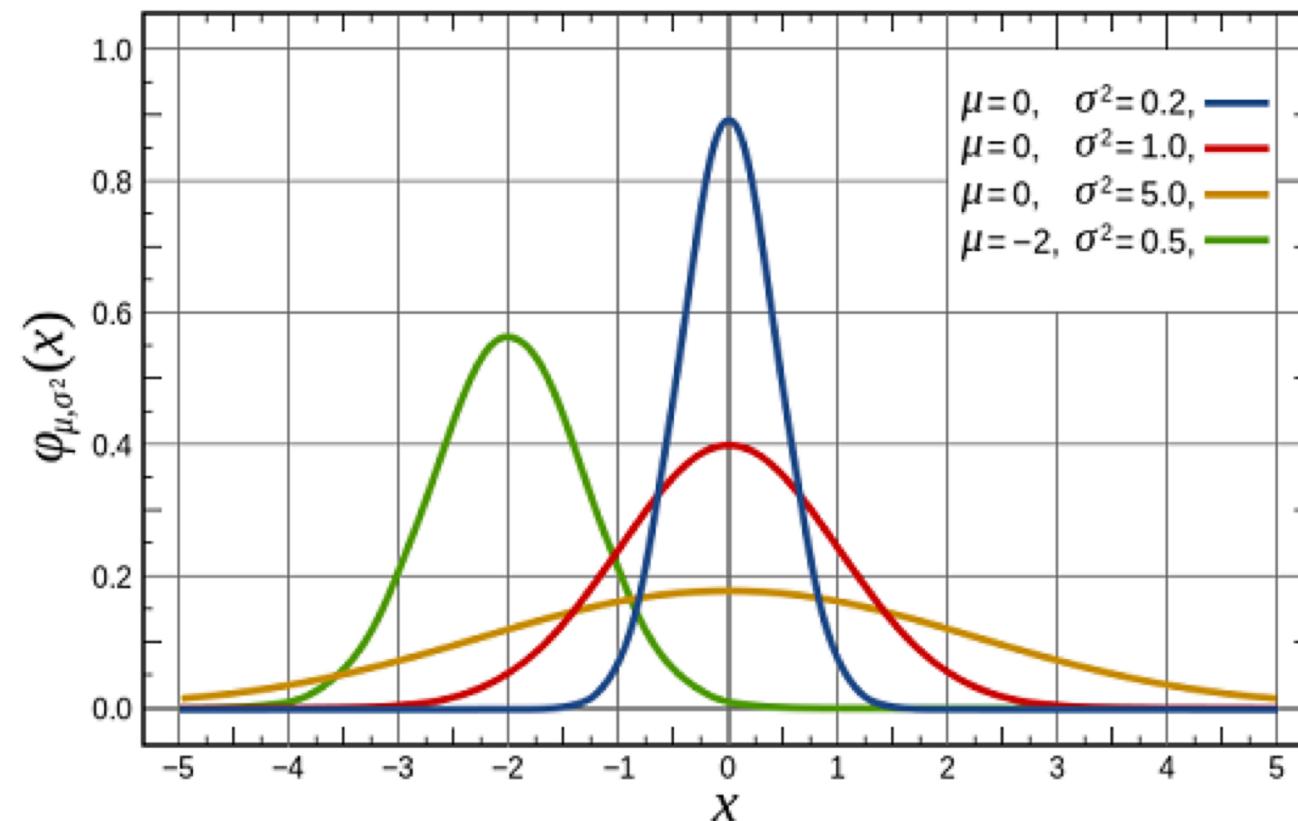


$$\frac{1}{6} \times (1 + 2 + 3 + 4 + 5 + 6) = 3.5$$

$$E_{x \sim P(X)}[reward(x)] = \sum_{x=1}^6 P(X = x) \times reward(x) = 3.5$$

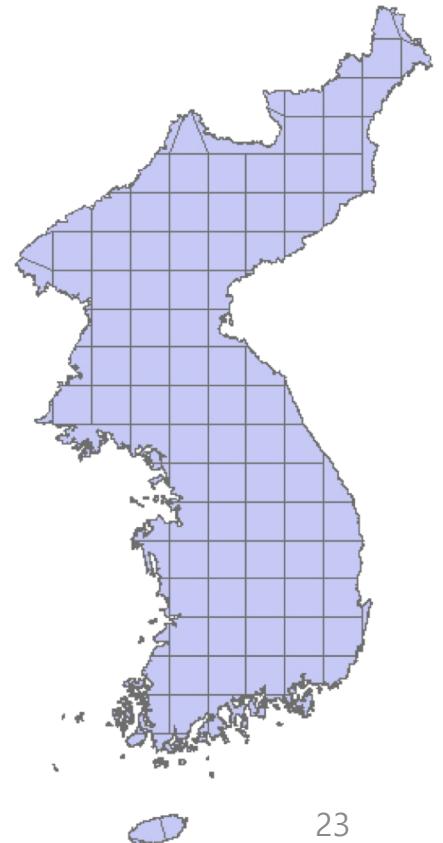
→ 평균 보상

Expectation



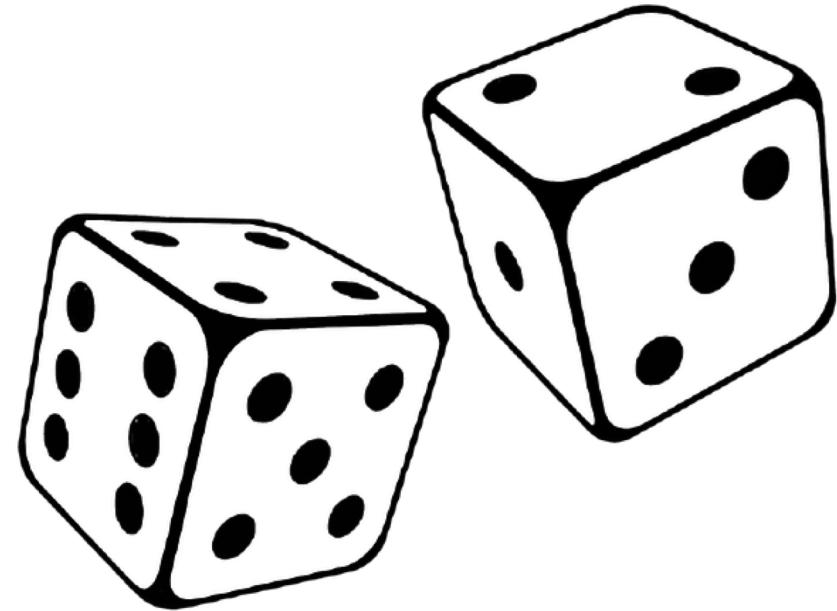
Monte Carlo Sampling

- Random sampling을 통해서 임의의 함수를 approximation.
 - 주사위가 정육각체가 아닐 경우 어떻게 근사할 것인가?
- 2차원 공간 상의 한반도의 넓이를 근사 해보자!



Monte Carlo Sampling

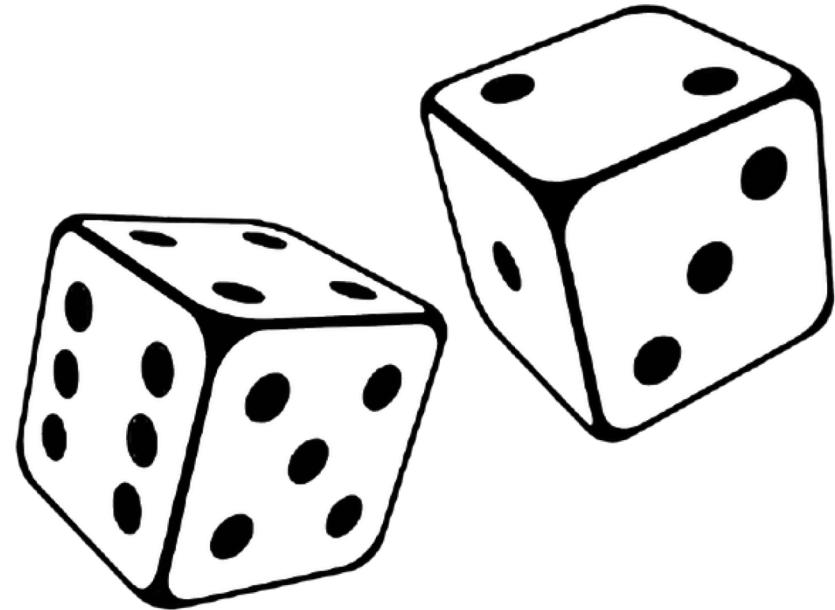
- Expected reward from dice
 - Try N-times and get average rewards
 - When N is getting bigger, approximation would close to real value.
 - Thus, we can write like as below:



$$E_{x \sim P(X)}[\text{reward}(x)] \approx \frac{1}{N} \sum_{i=1}^N \text{reward}(x_i)$$

Monte Carlo Sampling

- What if N equals to 1?
 - What if we get a just one sample?



$$E_{x \sim P(X)}[\text{reward}(x)] \approx \text{reward}(x) = x$$

Expectation

- What if dice has non-discrete surface?

$$\begin{aligned}\mathbb{E}_{x \sim p}[\text{reward}(x)] &= \int \text{reward}(x)p(x)dx \\ &\approx \frac{1}{K} \sum_{i=1}^K \text{reward}(x_i) \\ &\approx \text{reward}(x)\end{aligned}$$

Index

- Basics
 - Probability
 - Expectation and Sampling
 - Machine Learning
 - Information
 - Gradient Based Optimization
- Natural Language Processing
 - What is NLP?
 - History of Deep Learning
 - Why NLP is Hard?
 - Why Korean NLP is Hell?
 - Recent Trends of NLP

Machine Learning

- Objective:
 - Generalization: Have a good prediction for unseen data.

Machine Learning

$$\underbrace{P(Y|X)}_{posterior} = \frac{\overbrace{P(X|Y)P(Y)}^{likelihood \ prior}}{\underbrace{P(X)}_{evidence}}$$

수식	영어 명칭	한글 명칭
$P(Y X)$	Posterior	사후 확률
$P(X Y)$	Likelihood	가능도(우도)
$P(Y)$	Prior	사전 확률
$P(X)$	Evidence	증거

Maximum Likelihood Estimation (MLE)

$$\hat{\theta} = \operatorname{argmax}_{\theta} P(Y|X; \theta) = \operatorname{argmax}_{\theta} P(Y|X, \theta)$$

$$\hat{\theta} = \operatorname{argmax}_{\theta} P(X; \theta) = \operatorname{argmax}_{\theta} P(X|\theta)$$

MLE Example

- Predict a probability that flat-side goes to bottom.



MLE Example

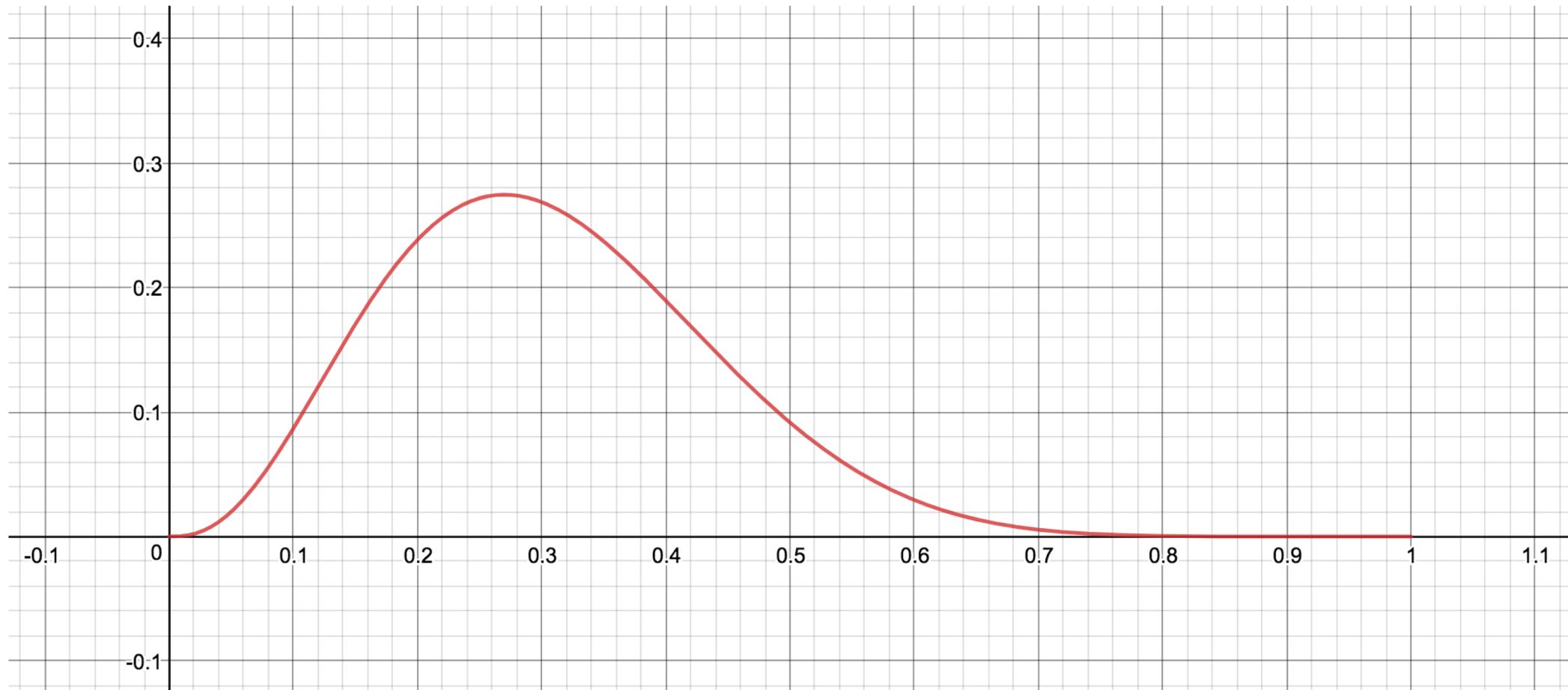
$$K \sim \mathcal{B}(n, \theta)$$

$$\begin{aligned} P(K = k) &= \binom{n}{k} \theta^k (1 - \theta)^{n-k} \\ &= \frac{n!}{k!(n - k)!} \cdot \theta^k (1 - \theta)^{n-k} \end{aligned}$$

MLE Example

- $k=27$
- $n=100$

MLE Example



Maximum A Posterior (MAP)

$$\begin{aligned}\hat{\theta} &= \operatorname{argmax}_{\theta} P(\theta|X, Y) \\&= \operatorname{argmax}_{\theta} P(X, Y, \theta) \\&= \operatorname{argmax}_{\theta} P(Y|X; \theta)P(X, \theta) \\&= \operatorname{argmax}_{\theta} P(Y|X; \theta)P(X; \theta)P(\theta)\end{aligned}$$

$$\begin{aligned}\hat{\theta} &= \operatorname{argmax}_{\theta} P(\theta|X) \\&= \operatorname{argmax}_{\theta} P(X, \theta) \\&= \operatorname{argmax}_{\theta} P(X; \theta)P(\theta)\end{aligned}$$

Ensemble?

$$P(Y|X) = \mathbb{E}_{\theta \sim P}[P(Y|X; \theta)] \approx \frac{1}{N} \sum_{i=1}^N P(Y|X; \theta)$$

Index

- Basics
 - Probability
 - Expectation and Sampling
 - Machine Learning
 - Information
 - Gradient Based Optimization
- Natural Language Processing
 - What is NLP?
 - History of Deep Learning
 - Why NLP is Hard?
 - Why Korean NLP is Hell?
 - Recent Trends of NLP

Information

- 정보이론에서 엔트로피는 어떤 정보의 불확실성을 나타냄
- 불확실성은 일어날 것 같은 사건(likely event)의 확률
 - 자주 발생하는(일어날 확률이 높은) 사건은 낮은 정보량을 가진다.
 - 드물게 발생하는(일어날 확률이 낮은) 사건은 높은 정보량을 가진다.
- 불확실성 $\propto 1/\text{확률} \propto \text{정보량}$

Information

- ① 내일 아침에는 해가 동쪽에서 뜬다.
 - ② 내일 아침에는 해가 서쪽에서 뜬다.
-
- a. 대한민국 올 여름의 평균 기온은 섭씨 28도로 예상 된다.
 - b. 대한민국 올 여름의 평균 기온은 섭씨 5도로 예상 된다.

Information

- 정보량
 - $-\log$ 때문에, 확률이 0에 가까워질수록 높은 정보량

$$I(x) = -\log P(x)$$

Entropy

- 정보량의 기대값(expectation)

$$H(P) = -E_{X \sim P}[\log P(x)] = -\sum_{\forall x} P(x) \log P(x)$$

Cross-Entropy

- Entropy of Q based on samplings from P.

$$H(P, Q) = -E_{X \sim P}[\log Q(x)] = -\sum_{\forall x} P(x) \log Q(x)$$

KL Divergence

$$\begin{aligned} KL(P||P_\theta) &= -\mathbb{E}_{X \sim P} [\log \frac{P_\theta(X)}{P(X)}] \\ &= -\sum_{x \in \mathcal{X}} P(x) \log \frac{P_\theta(x)}{P(x)} \\ &= -\sum_{x \in \mathcal{X}} \left(P(x) \log P_\theta(x) - P(x) \log P(x) \right) \\ &= H(P, P_\theta) - H(P) \end{aligned}$$

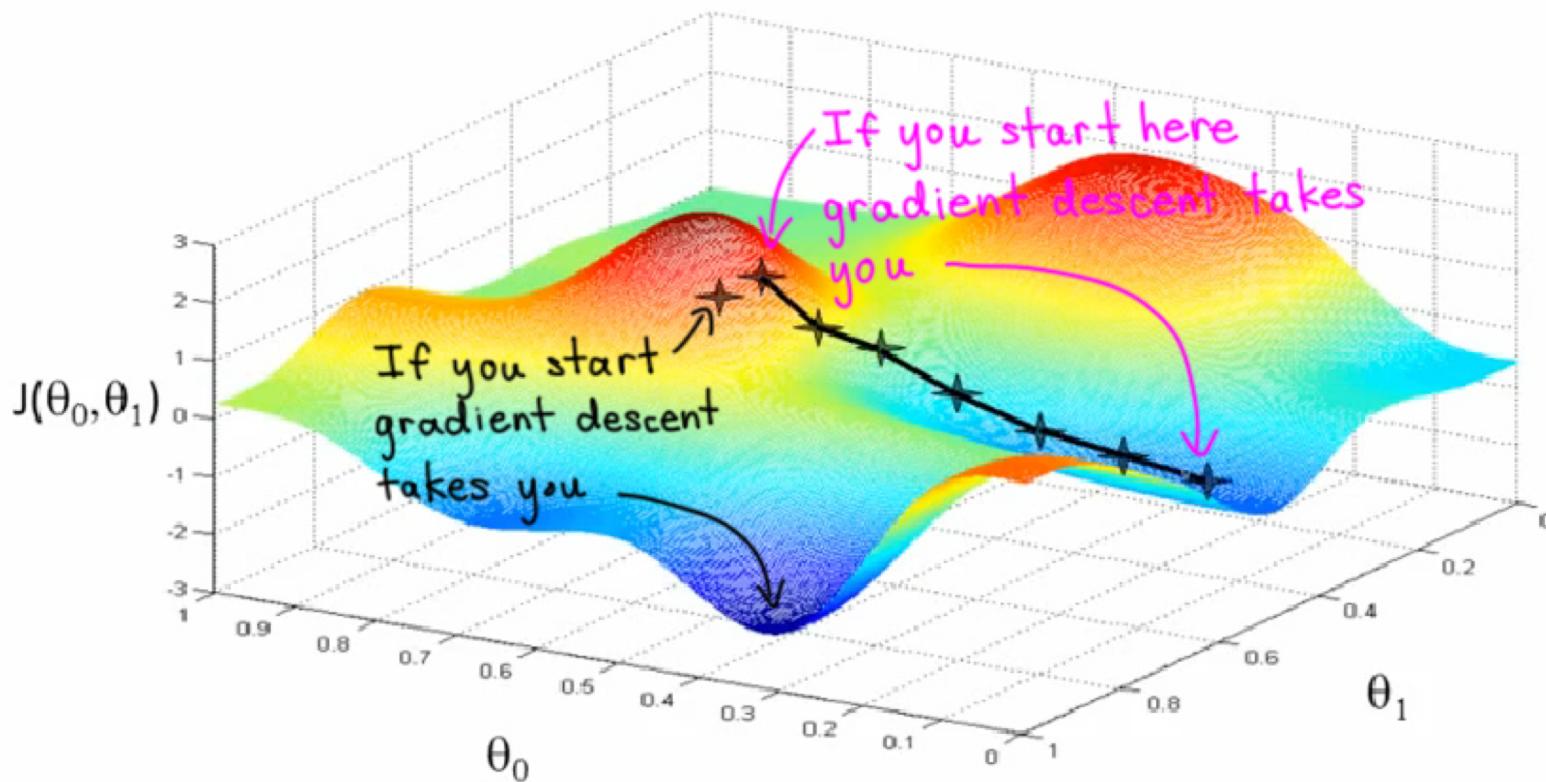
KL Divergence

$$\begin{aligned}\nabla_{\theta} KL(P||P_{\theta}) &= \nabla_{\theta} \left(H(P, P_{\theta}) - H(P) \right) \\ &= \nabla_{\theta} H(P, P_{\theta})\end{aligned}$$

Index

- Basics
 - Probability
 - Expectation and Sampling
 - Machine Learning
 - Information
 - Gradient Based Optimization
- Natural Language Processing
 - What is NLP?
 - History of Deep Learning
 - Why NLP is Hard?
 - Why Korean NLP is Hell?
 - Recent Trends of NLP

Gradient Descent



Gradient Descent

An objective function by Cross Entropy is

$$\begin{aligned} J(\theta) = H(P, P_\theta) &= -\mathbb{E}_{X \sim P(X)} \left[\mathbb{E}_{Y \sim P(Y|X)} [\log P(Y|X; \theta)] \right] \\ &= - \sum_{x \in \mathcal{X}} P(x) \sum_{y \in \mathcal{Y}} P(y|x) \log P(y|x; \theta) \end{aligned}$$

Gradient Descent

By Monte-Carlo Sampling,

$$\mathcal{B} = \{x, y\}_{i=1}^N$$

$$\begin{aligned} J(\theta) &\approx -\frac{1}{N} \sum_{i=1}^N \frac{1}{K} \sum_{j=1}^K \log P(y_j|x_i; \theta) \\ &\approx -\frac{1}{N} \sum_{i=1}^N \log P(y|x_i; \theta) \end{aligned}$$

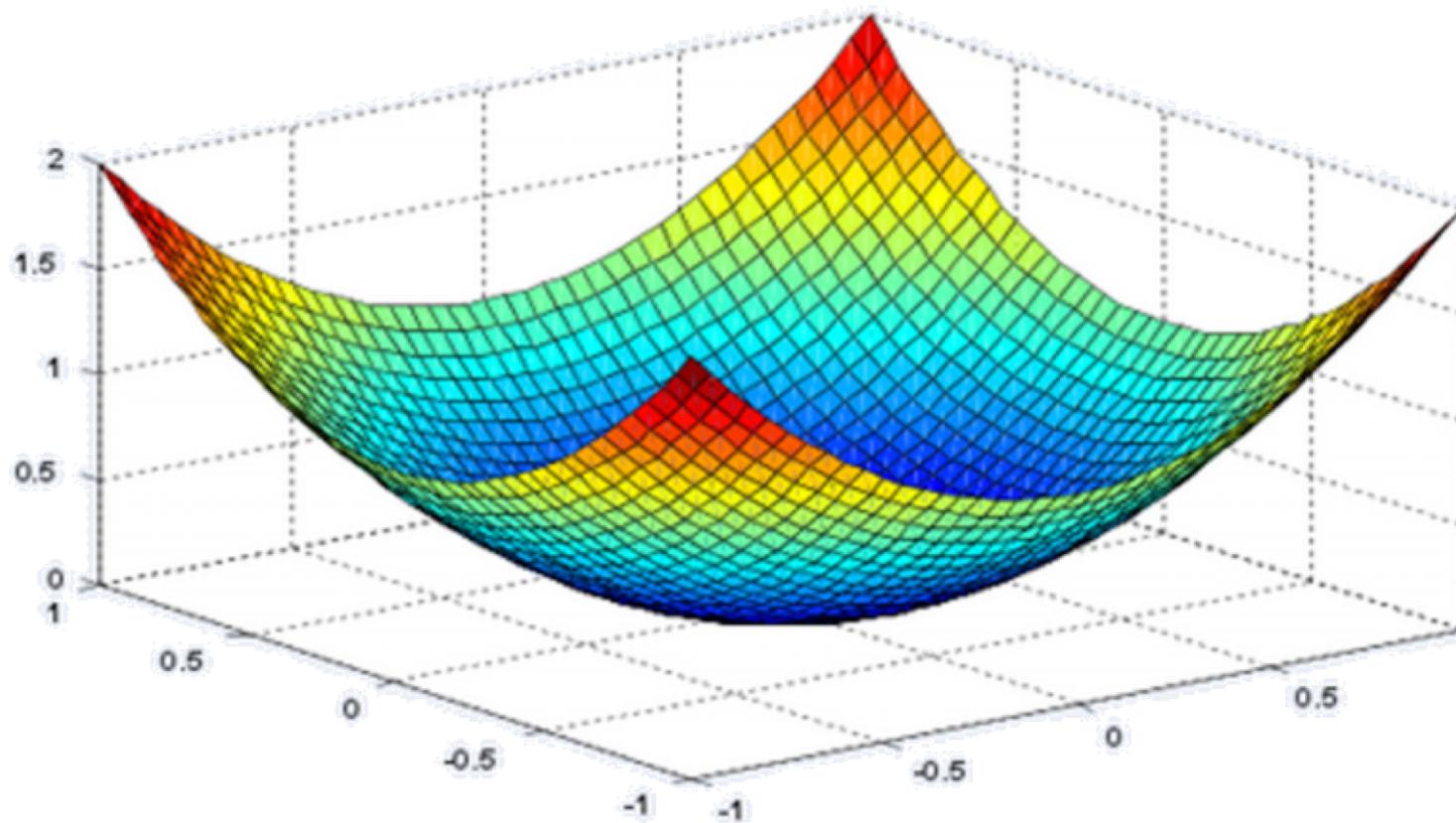
Gradient Descent

To minimize the objective function,

$$\hat{\theta} = \operatorname{argmin}_{\theta} J(\theta)$$

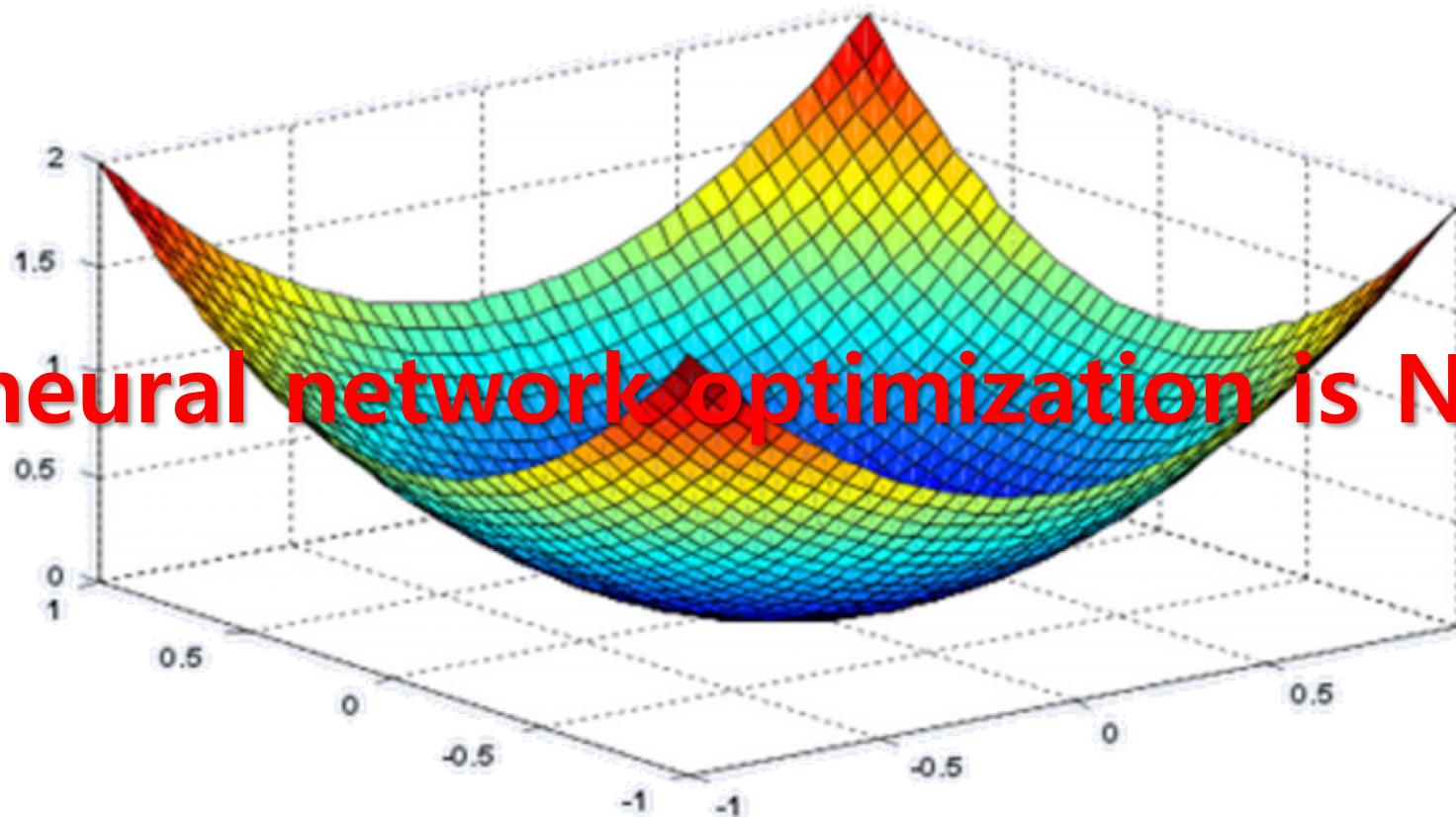
$$\theta \leftarrow \theta - \lambda \nabla_{\theta} J(\theta)$$

Convex Optimization



Convex Optimization

However, neural network optimization is NOT convex



Appendix: MSE Loss

- Assume that the distribution is in continuous space.

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

$$f_\theta(x) = \mathcal{N}(\mu_\phi(x), \sigma_\psi(x)^2)$$

Appendix: MSE Loss

$$\begin{aligned} J(\theta) &= -\frac{1}{N} \sum_{i=1}^N \log f_\theta(x_i) = -\frac{1}{N} \sum_{i=1}^N \log \left(\frac{1}{\sigma_\psi(x_i)\sqrt{2\pi}} \exp \left(-\frac{(x_i - \mu_\phi(x_i))^2}{2\sigma_\psi(x_i)^2} \right) \right), \\ &= -\frac{1}{N} \sum_{i=1}^N \left(\log \frac{1}{\sigma_\psi(x_i)\sqrt{2\pi}} - \frac{(x_i - \mu_\phi(x_i))^2}{2\sigma_\psi(x_i)^2} \right) \\ &= \log \sigma_\psi(x_i) + \frac{1}{2} \log 2\pi + \frac{1}{2\sigma_\psi(x_i)^2 \cdot N} \sum_{i=1}^N (x_i - \mu_\phi(x_i))^2 \end{aligned}$$

where $\theta^* = \{\phi, \psi\}$, but ignore ψ and $\theta = \{\phi\}$.

Appendix: MSE Loss

$$\hat{\theta} = \operatorname{argmin}_{\theta} J(\theta)$$

$$\theta \leftarrow \theta - \lambda \nabla_{\theta} J(\theta)$$

$$\nabla_{\theta} J(\theta) = \nabla_{\theta} \frac{1}{2\sigma^2 \cdot N} \sum_{i=1}^N (x_i - \mu_{\phi}(x_i))^2$$

$$\tilde{J}(\theta) = \frac{1}{N} \sum_{i=1}^N (x_i - \mu_{\phi}(x_i))^2$$

References

- Deep Learning Book [\[Goodfellow et al.2016\]](#)

Index

- Basics
 - Probability
 - Expectation and Sampling
 - Machine Learning
 - Information
 - Gradient Based Optimization
- Natural Language Processing
 - What is NLP?
 - History of Deep Learning
 - Why NLP is Hard?
 - Why Korean NLP is Hell?
 - Recent Trends of NLP

What is NLP?

- 인공지능 분야의 큰 branch 중 하나
 - 인공지능 삼대장
 - Computer Vision – Image Recognition
 - Automatic Speech Recognition (ASR)
 - Natural Language Processing – Machine Translation
- **대용량의 Text Corpus를 사용/처리**하여 컴퓨터에게 사람이 사용하는 언어를 처리하고 이해하도록 함
 - 사람과 컴퓨터 사이의 **매개체, 인터페이스**



What is NLP?

- Linguistics와 **융합** 학문
 - Traditional Linguistics
 - 언어 자체의 비밀을 밝혀 내는 것이 목적
 - 세세한 규칙 완벽하게 언어를 묘사하는 하나하나까지 관심을 가짐
 - Computational Linguistics
 - 언어 자체의 아우르는 규칙을 찾아내는 것이 목적
 - 최대한 넓은 coverage를 갖는 규칙을 찾자!
- Computational Linguistics를 통해 Natural Language Processing을 구현(engineering)

Sub-Topics on NLP

- Phonetics and Phonology, 음운론
 - the study of linguistic sounds
- Morphology , 형태론
 - the study of the meaning of components of words
- Syntax , 구문론
 - the study of the structural relationships between words
- Semantics , 의미론
 - the study of meaning
- Discourse , 담론
 - they study of linguistic units larger than a single utterance

Applications on NLP

- 최종목표: **사람의 언어를 이해**하여 컴퓨터로 하여금 여러가지 tasks를 수행할 수 있도록 하는 것
- Siri, Alexa와 같이 사용자의 의도를 파악하고 대화하거나 도움을 주는 task
- 요약, 번역과 같은 task
- 감성분석과 같이 대량의 텍스트를 이해하고 수치화 하는 task
- 사용자로부터 입력을 받아 사용자가 원하는 것을 검색 및 답변을 주는 task

Index

- Basics
 - Probability
 - Expectation and Sampling
 - Machine Learning
 - Information
 - Gradient Based Optimization
- Natural Language Processing
 - What is NLP?
 - History of Deep Learning
 - Why NLP is Hard?
 - Why Korean NLP is Hell?
 - Recent Trends of NLP

Dawn

- Before 2010's

Table 1: Major milestones that will be covered in this paper

Year	Contributer	Contribution
300 BC	Aristotle	introduced Associationism, started the history of human's attempt to understand brain.
1873	Alexander Bain	introduced Neural Groupings as the earliest models of neural network, inspired Hebbian Learning Rule.
1943	McCulloch & Pitts	introduced MCP Model, which is considered as the ancestor of Artificial Neural Model.
1949	Donald Hebb	considered as the father of neural networks, introduced Hebbian Learning Rule, which lays the foundation of modern neural network.
1958	Frank Rosenblatt	introduced the first perceptron, which highly resembles modern perceptron.
1974	Paul Werbos	introduced Backpropagation
1980	Teuvo Kohonen	introduced Self Organizing Map
	Kunihiko Fukushima	introduced Neocogitron, which inspired Convolutional Neural Network
1982	John Hopfield	introduced Hopfield Network
1985	Hilton & Sejnowski	introduced Boltzmann Machine
1986	Paul Smolensky	introduced Harmonium, which is later known as Restricted Boltzmann Machine
	Michael I. Jordan	defined and introduced Recurrent Neural Network
1990	Yann LeCun	introduced LeNet, showed the possibility of deep neural networks in practice
1997	Schuster & Paliwal	introduced Bidirectional Recurrent Neural Network
	Hochreiter & Schmidhuber	introduced LSTM, solved the problem of vanishing gradient in recurrent neural networks
2006	Geoffrey Hinton	introduced Deep Belief Networks, also introduced layer-wise pretraining technique, opened current deep learning era.
2009	Salakhutdinov & Hinton	introduced Deep Boltzmann Machines
2012	Geoffrey Hinton	introduced Dropout, an efficient way of training neural networks

Image Recognition

- AlexNet [[Krizhevsky et al.2012](#)] 이미지넷 우승 (2012)
 - 딥러닝의 시대의 서막
 - 여러 층의 Convolutional Layer을 쌓아서 architecture를 만듦
 - 당시 3GB 메모리의 Nvidia GTX580을 2개 사용하여 훈련
- 이후, ImageNet은 딥러닝의 경연장
- 결국, ResNet([He et al.2015](#))은 Residual Connection을 활용하여 150층이 넘는 deep architecture를 구성하며 우승

Image Recognition

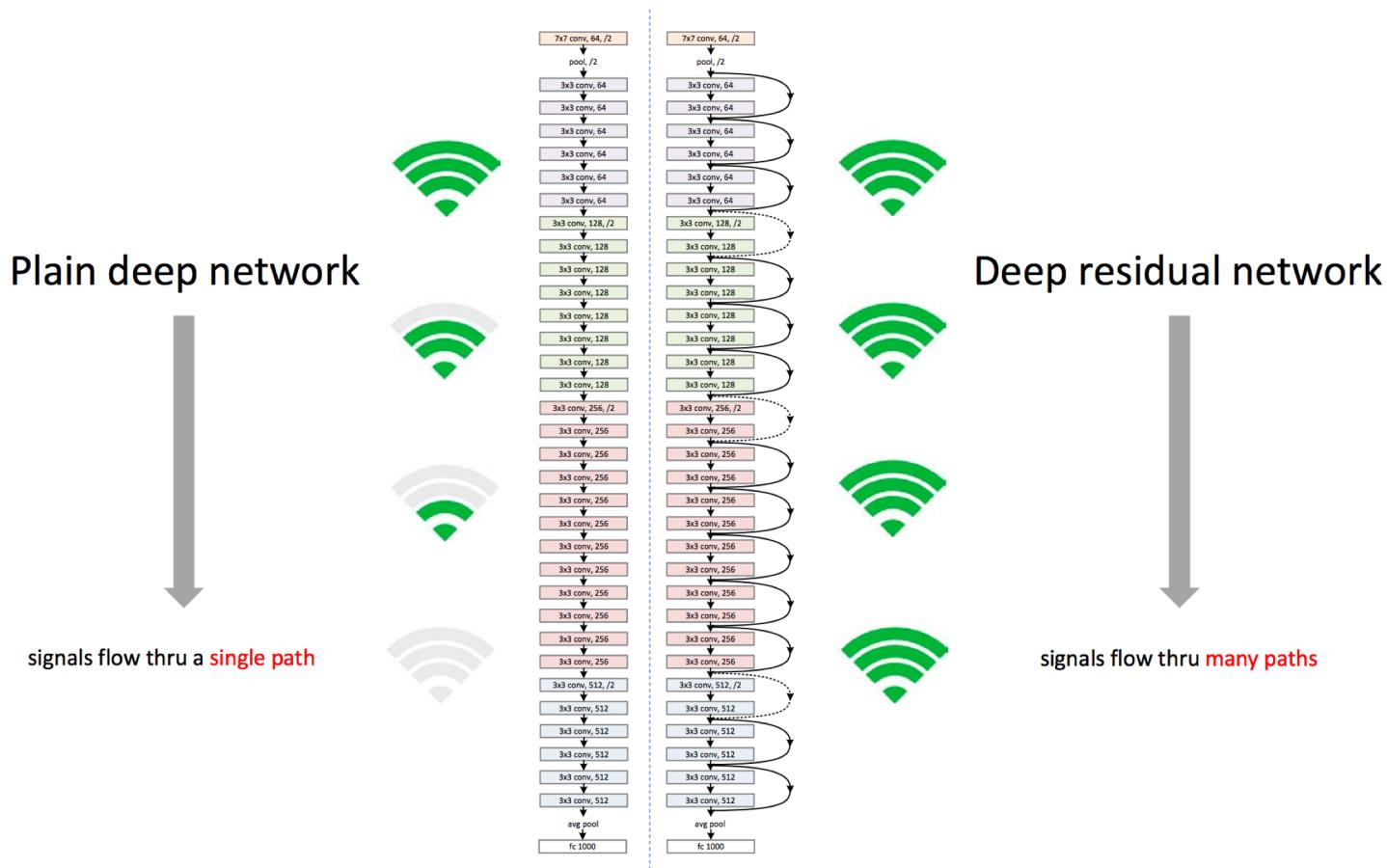
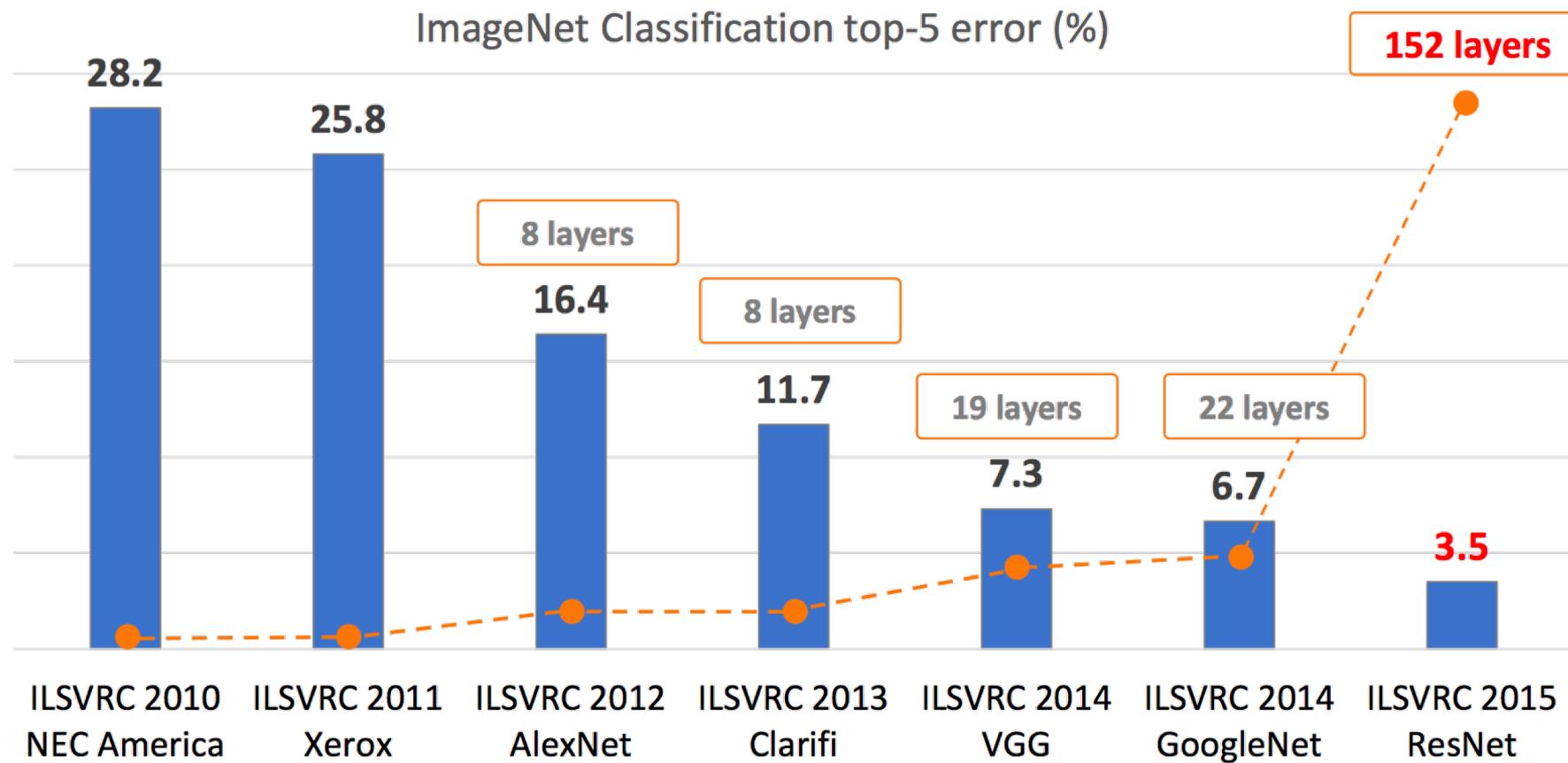


Image Recognition



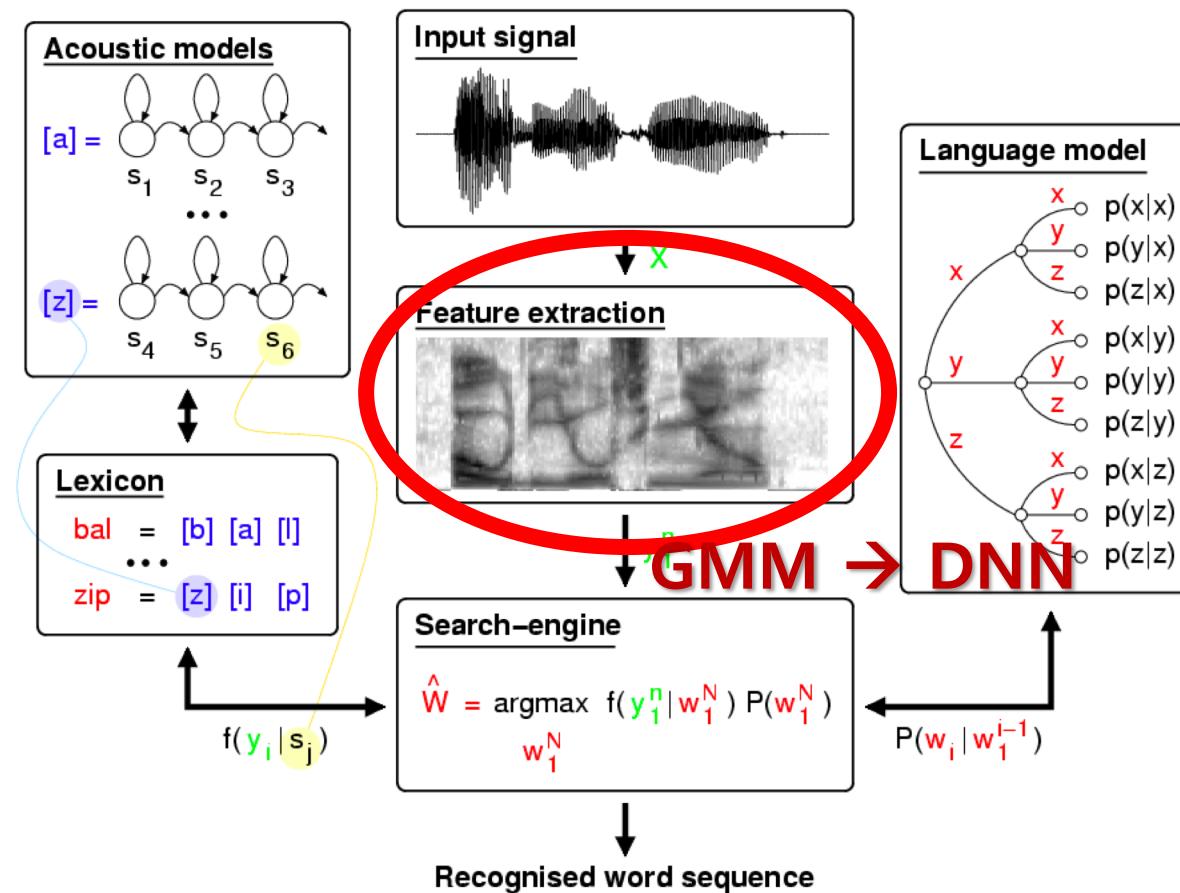
Speech Recognition

- 2000년대에 들어 큰 정체기
- Acoustic Model (AM)
 - GMM(Gaussian Mixture Model)을 통해 phone을 인식
 - HMM(Hidden Markov Model)을 통해 sequential하게 modeling
- Language Model (LM)
 - n-gram기반
- WFST(Weighted Finite State Transducer)방식을 통해 결합
- 너무나도 복잡한 구조와 함께 그 성능의 한계

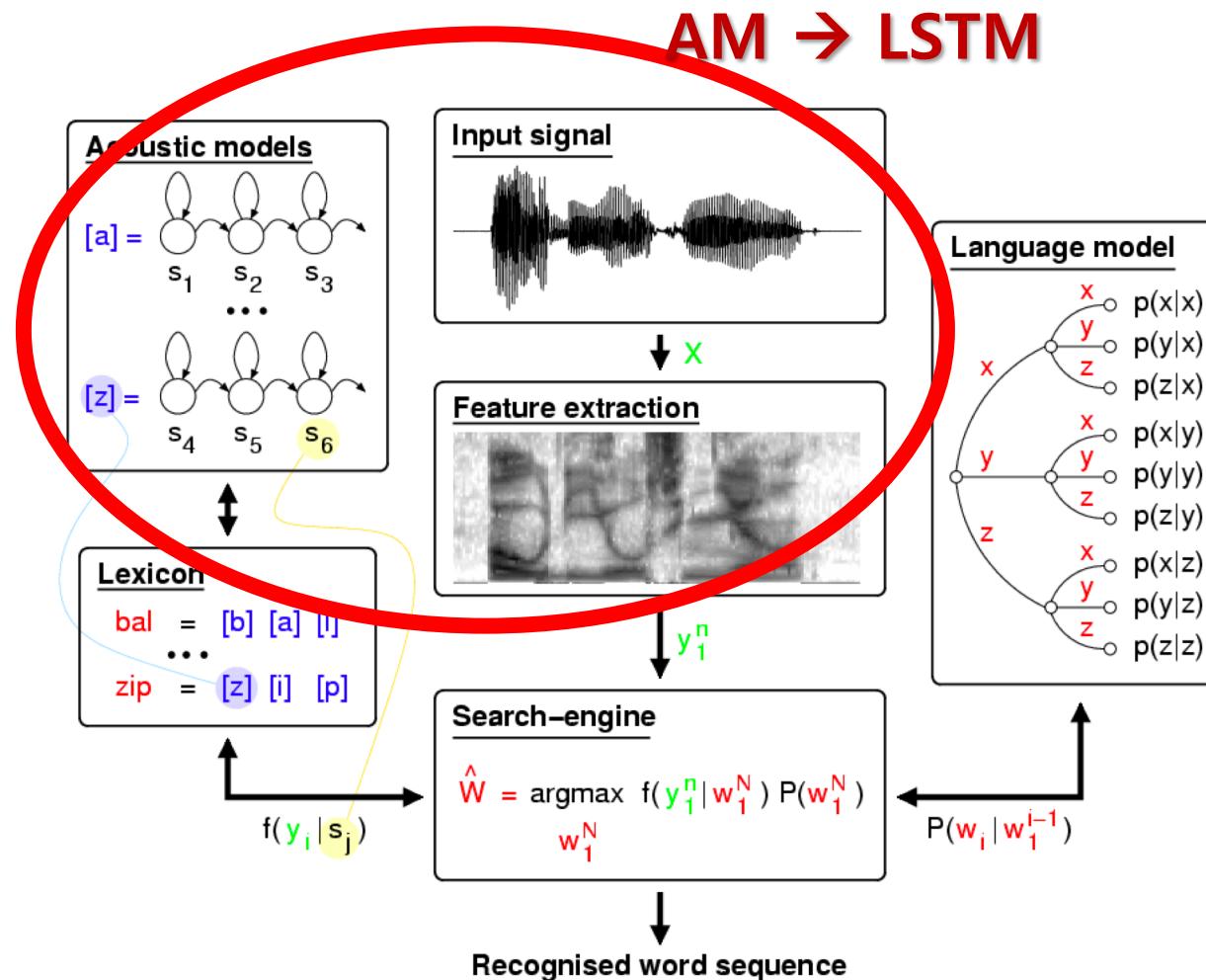
Speech Recognition

- 딥러닝(당시에는 Deep Neural Network라고 불림)을 활용하여 큰 발전
 - 2012년 GMM을 DNN으로 대체
 - 십 수년간의 정체를 단숨에 뛰어넘는 큰 혁명
 - 점차 AM전체를 LSTM으로 대체
 - end-to-end model([[Chiu et al.2017](#)])이 점점 확대되는 추세
- 오히려 이 분야에서는 vision분야에 비해서 딥러닝 기술을 활용하여 상용화에까지 성공한 더욱 인상적인 사례

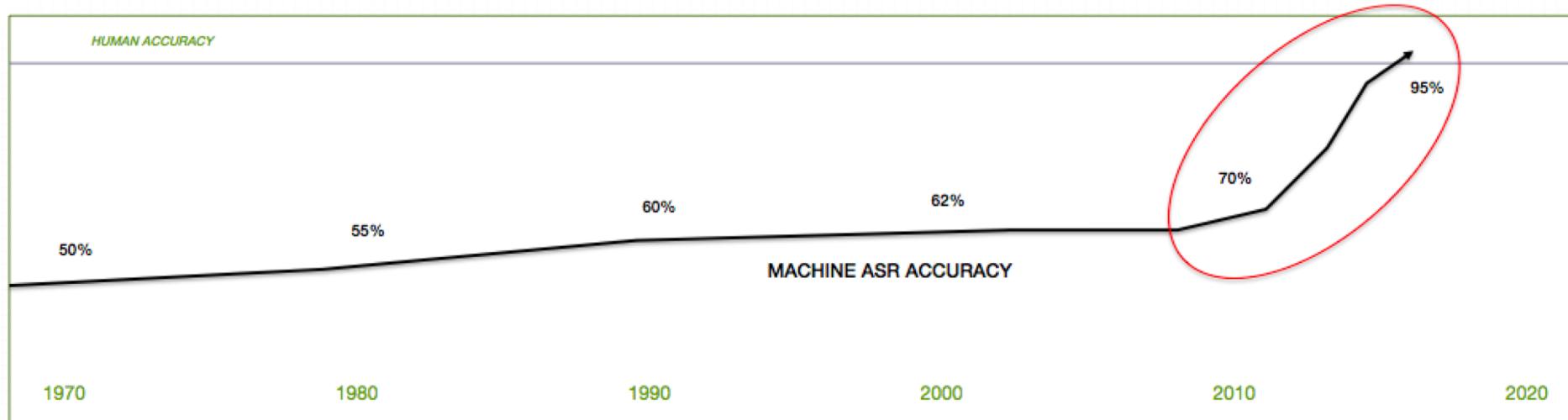
Speech Recognition



Speech Recognition



Speech Recognition

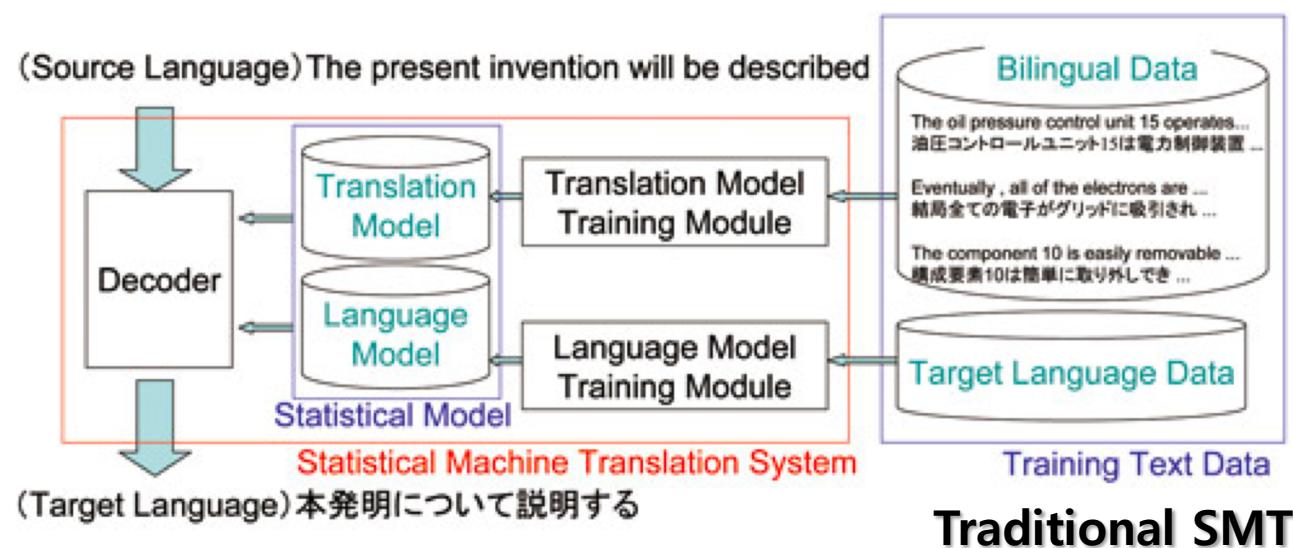


But...

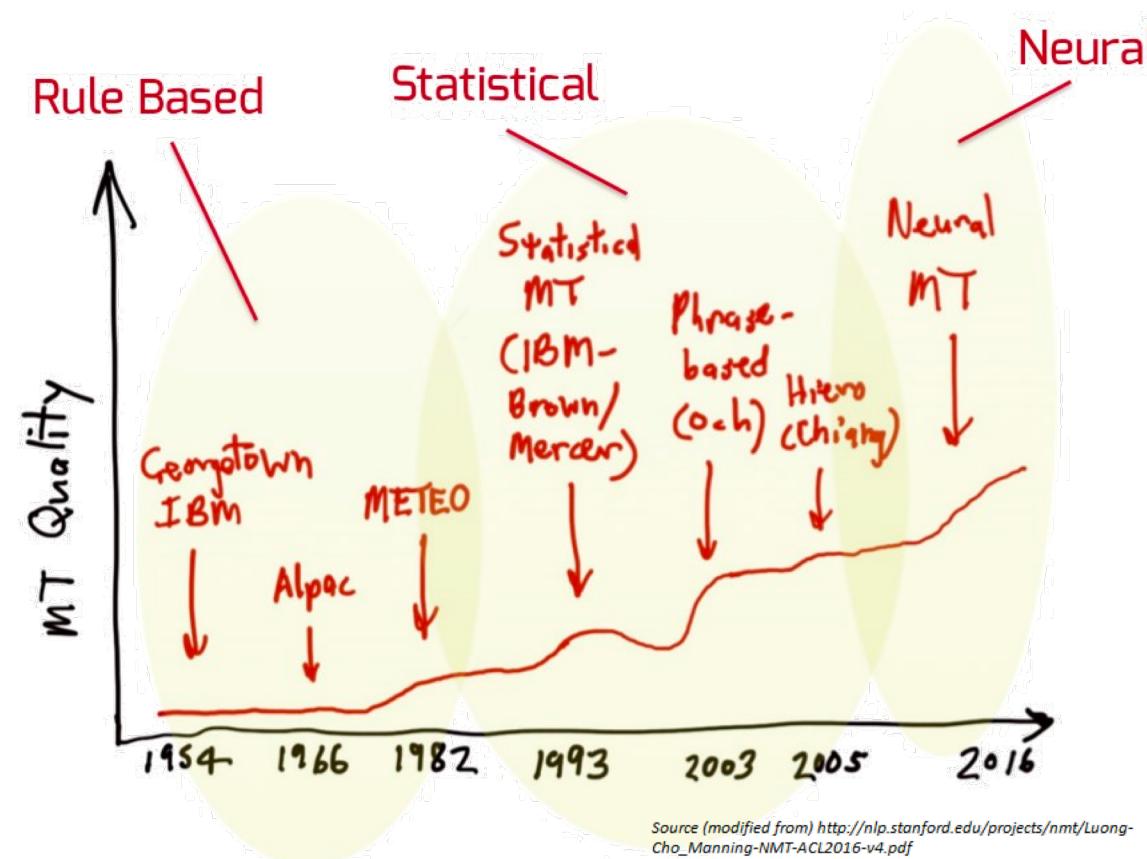


Neural Machine Translation

- 2014년 Sequence-to-sequence(seq2seq)가 소개
 - end-to-end neural machine translation의 시대
- Natural Language Generation의 시대가 도래



Neural Machine Translation



Neural Machine Translation

- 결국, 기계번역은 가장 늦게 혁명이 이루어졌지만, 가장 먼저 딥러닝만을 사용해 상용화가 된 분야
- 현재의 상용 기계번역 시스템은 모두 딥러닝으로 대체.
- 읽을거리:
 - <https://devblogs.nvidia.com/introduction-neural-machine-translation-with-gpus/>
 - <https://devblogs.nvidia.com/introduction-neural-machine-translation-gpus-part-2/>
 - <https://devblogs.nvidia.com/introduction-neural-machine-translation-gpus-part-3/>

Neural Machine Translation

- 추가 읽을거리:
 - <http://newspeppermint.com/2016/12/31/ai-awakening/>
 - 10부까지 있음

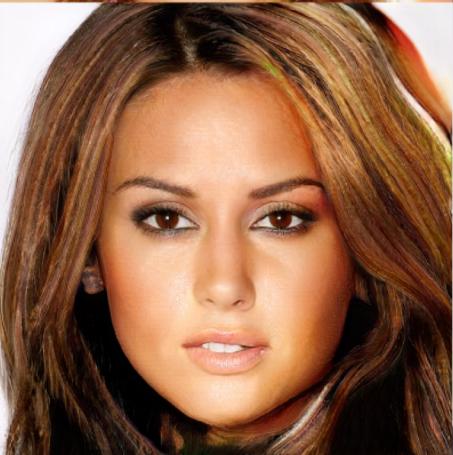
Generative Learning

- Neural Network은 **pattern classification**에 있어서 타 알고리즘에 비해서 너무나도 **압도적인** 성능
 - image recognition, text classification과 같은 단순한 분류 문제 (classification or discriminative learning)는 금방 정복
- Discriminative Learning $\hat{\theta} = \operatorname{argmax}_{\theta} P(y|x; \theta)$
- Generative Learning $\hat{\theta} = \operatorname{argmax}_{\theta} P(x; \theta)$

Generative Learning

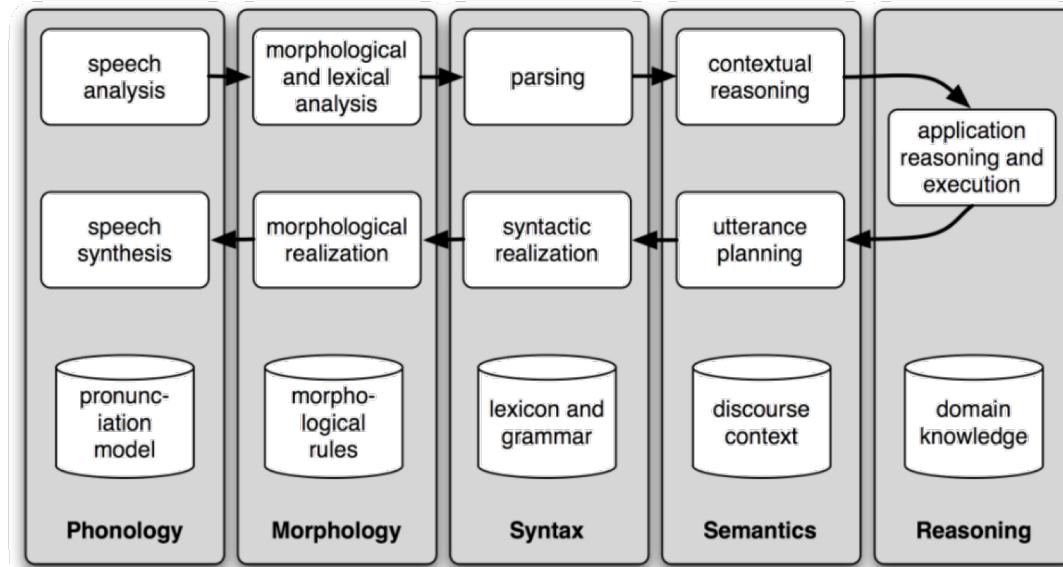
- Adversarial learning (GAN, [[Goodfellow et al.2014](#)])
- Variational Auto-encoder (VAE, [[Kingma et al.2013](#)])

Generative Learning



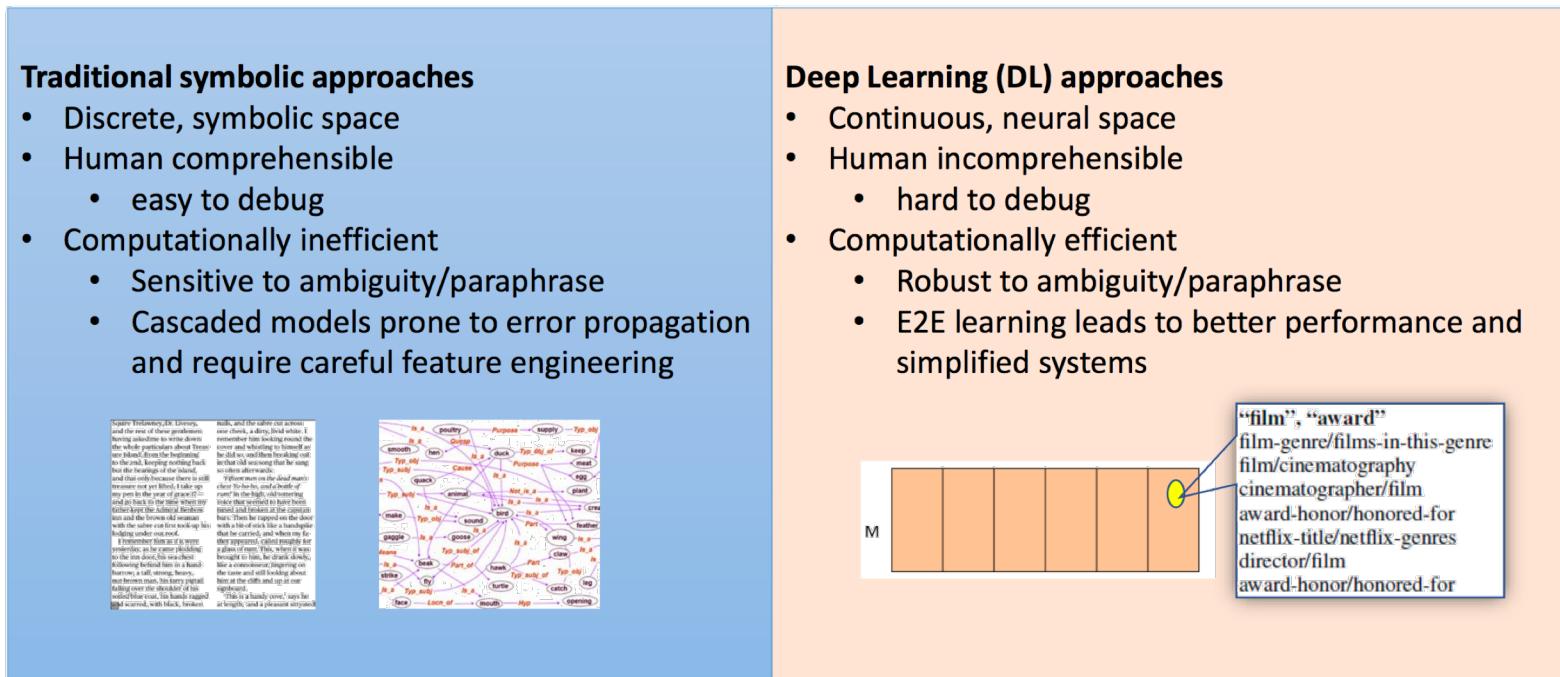
Paradigm Shift on NLP from Traditional to Deep Learning

- 전형적인 NLP application의 구조
 - 여러 단계의 sub-module로 구성되어 복잡한 디자인을 구성
 - 매우 무겁고 복잡하여 구현 및 시스템 구성이 어려운 단점
 - 각기 발생한 error가 중첩 및 가중되어 error propagation



Paradigm Shift on NLP from Traditional to Deep Learning

- 기계번역의 사례처럼 NLP 전반에 걸쳐 deep learning의 물결
- 처음에는 각 sub-module을 대체하는 형태로 진행
- 점차 기계번역의 사례처럼 결국 **end-to-end model들**로 대체



접근 방법의 변화

- Traditional NLP
 - 사람의 언어는 Discrete한 symbol
 - symbol간에는 유사성이 있을 수 있지만 모든 단어는 서로 다른 symbol
 - 전통적인 NLP에서는 discrete symbol로써 데이터를 취급
 - 사람이 데이터를 보고 해석하기는 쉬운 장점
 - 모호성이나 유의성을 다루는데 어려움
- NLP with Deep Learning
 - word embedding을 통해서 단어를 continuous한 vector로 나타냄
 - 모호성과 유의성에서도 이득
 - end-to-end model을 구현함으로써 더욱 높은 성능
 - RNN의 단점을 보완한 LSTM과 GRU에 대한 활용법이 고도화
 - Attention의 등장: 긴 time-step의 sequential 데이터에 대해서도 훈련

접근 방법의 변화

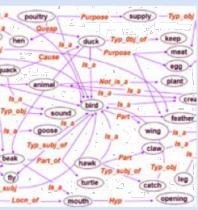
Symbolic Space

- **Knowledge Representation**
 - *Explicitly* store a BIG but incomplete knowledge graph (KG)
 - Words, relations, templates
 - High-dim, discrete, sparse vectors
- **Inference**
 - Slow on a big KG
 - Keyword/template matching is sensitive to paraphrase alternations
- **Human comprehensible but not computationally efficient**

Squire Falstaff, Dr. Greedy, and the rest of these geriatric bungling buffoons, who have done the whole particulars about their men, and the men themselves to the end; keeping nothing back but the bearings of the island, and the names of the towns, still treasure not yet lifted. I take up my pen again, and go back to the time when my father kept the Admiral Barbow inn, and the old soldiers, all decked out with the salve cut first took up his lodgings there.

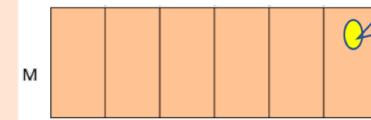
I remember him as if it were yesterday, as he came plodding into the room, a tall, gaunt, thin fellow behind him in a hand-barrow, with a small, dark brown man, his terry pajamal falling over the shoulder of his bedfellow, and the man, who was a ragged

old fellow, with black, beaded eyes,



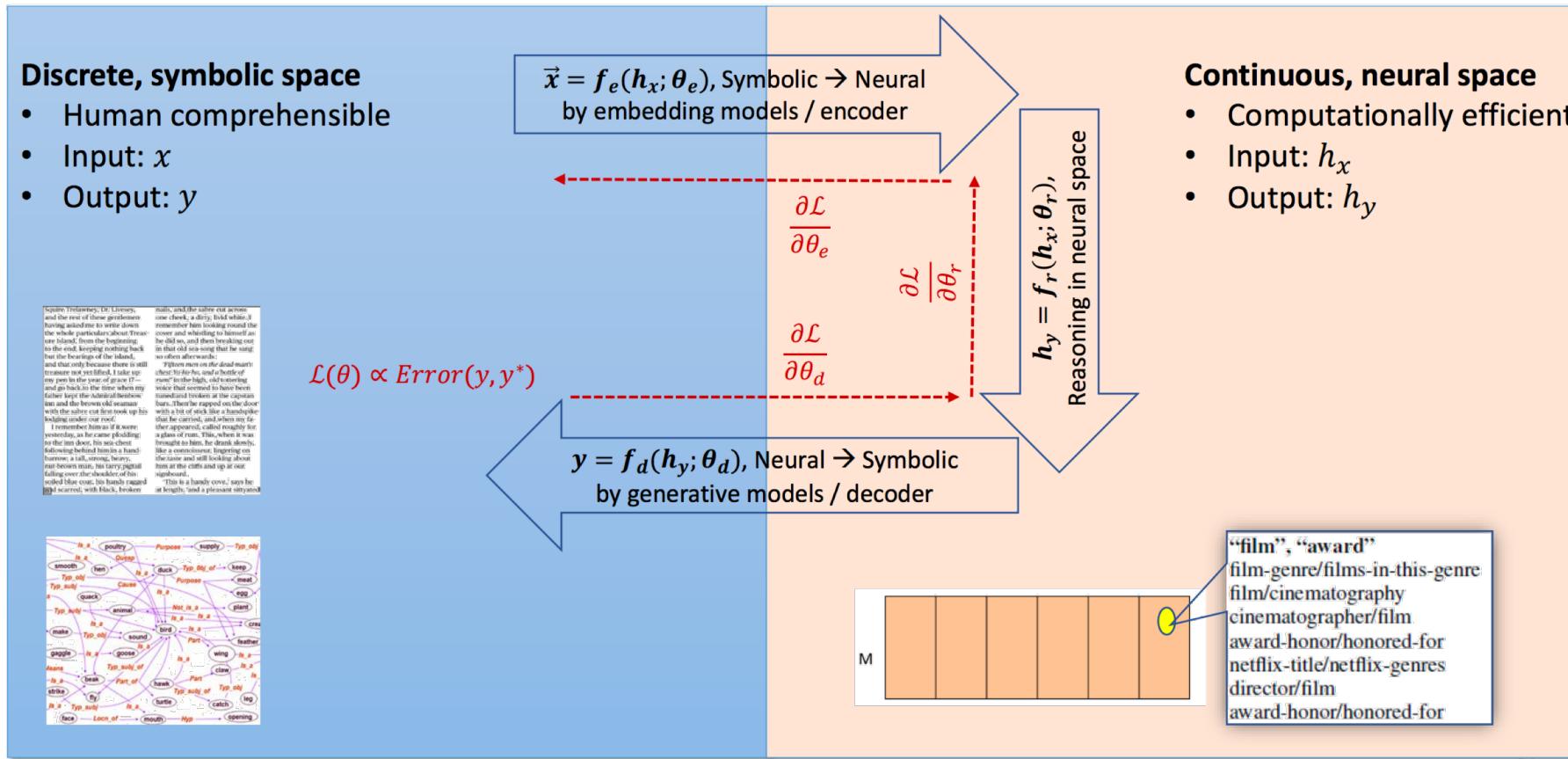
Neural Space

- **Knowledge Representation**
 - *Implicitly* store entities and structure of KG in a *compact* way that is **more generalizable**
 - Semantic concepts/classes
 - Low-dim, cont., dense vectors shaped by KG
- **Inference**
 - **Fast** on compact memory
 - Semantic matching is **robust** to paraphrase alternations
- **Computationally efficient but not human comprehensible yet**

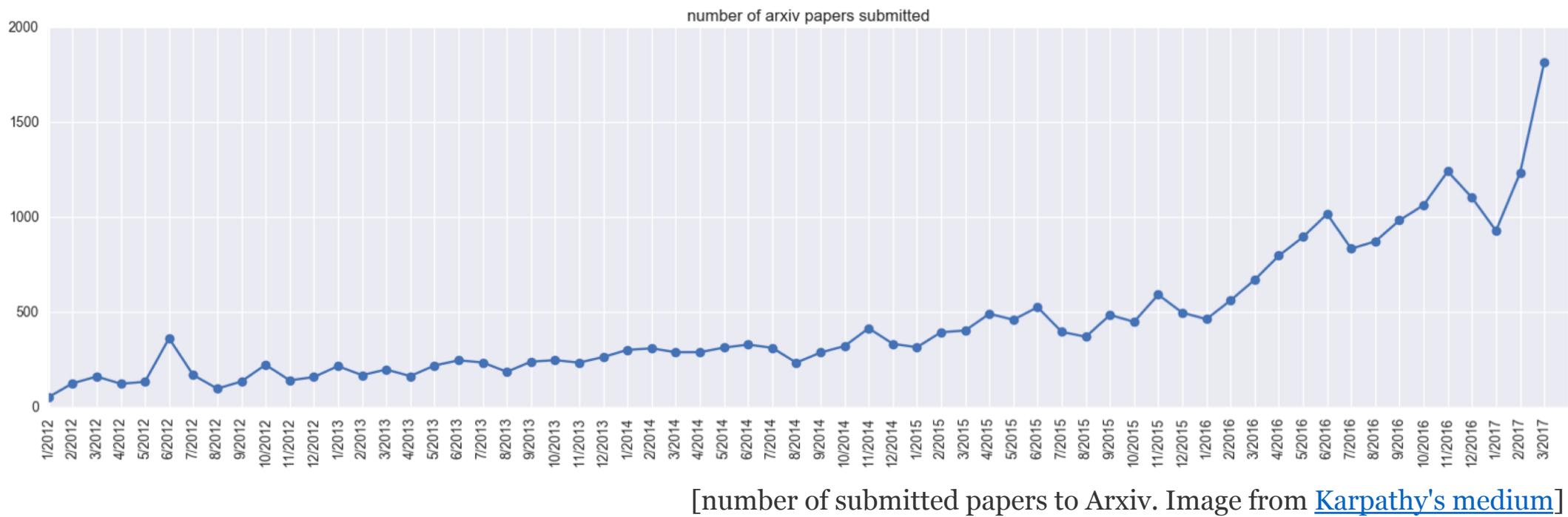


“film”, “award”
film-genre/films-in-this-genre
film/cinematography
cinematographer/film
award-honor/honored-for
netflix-title/netflix-genres
director/film
award-honor/honored-for

NLP System with Deep Learning



Peak of Deep Learning Era?



References

- <https://www.microsoft.com/en-us/research/wp-content/uploads/2017/07/dl-summer-school-2017.-Jianfeng-Gao.v2.pdf>
- Stanford CS224n Lecture

Index

- Basics
 - Probability
 - Expectation and Sampling
 - Machine Learning
 - Information
 - Gradient Based Optimization
- Natural Language Processing
 - What is NLP?
 - History of Deep Learning
 - Why NLP is Hard?
 - Why Korean NLP is Hell?
 - Recent Trends of NLP

Ambiguity

원문	차를 마시러 공원에 가던 차 안에서 나는 그녀에게 차였다.
G*	I was kicking her in the car that went to the park for tea.
M*	I was a car to her, in the car I had a car and went to the park.
N*	I got dumped by her on the way to the park for tea.
K*	I was in the car going to the park for tea and I was in her car.
S*	I got dumped by her in the car that was going to the park for a cup of tea.

Ambiguity

단어 중의성 해소(word sense disambiguation)

원문	차를 마시러 공원에 가던 차 안에서 나는 그녀에게 차였다.
G*	I was kicking her in the car that went to the park for tea.
M*	I was a car to her, in the car I had a car and went to the park.
N*	I got dumped by her on the way to the park for tea.
K*	I was in the car going to the park for tea and I was in her car.
S*	I got dumped by her in the car that was going to the park for a cup of tea.

Ambiguity

원문	나는 철수를 안 때렸다.
1	철수는 맞았지만, 때린 사람이 나는 아니다.
2	나는 누군가를 때렸지만, 그게 철수는 아니다.
3	나는 누군가를 때린 적도 없고, 철수도 맞은 적이 없다.

Ambiguity

문장 내 정보의 부족으로 인한 모호성이 발생

원문	나는 철수를 안 때렸다.
1	철수는 맞았지만, 때린 사람이 나는 아니다.
2	나는 누군가를 때렸지만, 그게 철수는 아니다.
3	나는 누군가를 때린 적도 없고, 철수도 맞은 적이 없다.

Ambiguity

원문	선생님은 울면서 돌아오는 우리를 위로 했다.
1	(선생님은 울면서) 돌아오는 우리를 위로 했다.
2	선생님은 (울면서 돌아오는 우리를) 위로 했다.

Ambiguity

문장 내 정보의 부족이 야기한 구조 해석의 문제

원문	선생님은 울면서 돌아오는 우리를 위로 했다.
1	(선생님은 울면서) 돌아오는 우리를 위로 했다.
2	선생님은 (울면서 돌아오는 우리를) 위로 했다.

Paraphrase



Paraphrase

번호	문장 표현
1	여자가 김치를 어떤 남자에게 집어 던지고 있다.
2	여자가 어떤 남자에게 김치로 때리고 있다.
3	여자가 김치로 싸대기를 날리고 있다.
4	여자가 배추 김치 한 포기로 남자를 때리고 있다.
5	여자가 김치를 사용해 남자를 때리고 있다.
6	남자가 여자에게 김치로 싸대기를 맞고 있다.
7	남자가 여자로부터 김치로 맞고 있다.

Paraphrase

**문장의 표현 형식은 다양하고,
비슷한 의미의 단어들이 존재하기 때문에
paraphrase의 문제가 존재**

번호	문장 표현
1	여자가 김치를 어떤 남자에게 집어 던지고 있다.
2	여자가 어떤 남자에게 김치로 때리고 있다.
3	여자가 김치로 싸대기를 날리고 있다.
4	여자가 배추 김치 한 포기로 남자를 때리고 있다.
5	여자가 김치를 사용해 남자를 때리고 있다.
6	남자가 여자에게 김치로 싸대기를 맞고 있다.
7	남자가 여자로부터 김치로 맞고 있다.

Discrete, Not Continuous

- 사실은 discrete하기 때문에 그동안 쉽다고 생각
- Neural network에 적용 위해 continuous한 값으로 바꿔야 함
- Word embedding이 그 역할 훌륭하게 수행하지만,
 - Neural network 상에서 여러가지 방법을 구현할 때에 제약이 존재
 - 애초에 continuous한 값이 아니었기 때문

Discrete, Not Continuous

- Curse of Dimensionality
 - Discrete한 데이터
 - 많은 종류의 단어를 표현하기 위해서는 엄청난 dimension이 필요
 - 각 단어를 discrete한 symbol로 다루었기 때문
 - 마치 vocabulary size = $|V|$ 만큼의 dimension이 있는 것이나 마찬가지
 - Sparseness를 해결하기 위해서 단어를 적절하게 segmentation하는 등 여러가지 노력이 필요
- 적절한 word embedding을 통해서 dimension reduction을 하여 이 문제를 해결

Noise and Normalization

- Noise를 signal로 부터 적절히 분리해 내는 일은 매우 중요
- 자칫 실수하면 data는 본래의 의미마저 같이 잃어버릴 수도
- 이러한 관점에서 NLP는 어려움
 - 특히, 다른 종류의 데이터에 비해서 데이터가 살짝 바뀌었을 때의 의미의 변화가 훨씬 크기 때문
 - 이미지에서 한 픽셀의 RGB값이 각각 0에서 255까지로 나타내어지고, 그 값중 하나의 수치가 1이 바뀌었다고 해도 해당 이미지의 의미는 변화가 없음
 - 하지만 단어는 살짝만 바뀌어도 문장의 의미가 완전히 다르게 변할 수도 있음
 - 띠어쓰기나 어순의 차이로 인한 정제의 이슈
 - 이러한 어려움을 다루고 해결하기 위한 방법을 **Preprocessing**에서 다룰 것

Index

- Basics
 - Probability
 - Expectation and Sampling
 - Machine Learning
 - Information
 - Gradient Based Optimization
- Natural Language Processing
 - What is NLP?
 - History of Deep Learning
 - Why NLP is Hard?
 - Why Korean NLP is Hell?
 - Recent Trends of NLP

교착어

종류	대표적 언어	특징
교착어	한국어, 일본어, 몽골어	어간에 접사가 붙어 단어를 이루고 의미와 문법적 기능이 정해짐
굴절어	라틴어, 독일어, 러시아어	단어의 형태가 변함으로써 문법적 기능이 정해짐
고립어	영어, 중국어	어순에 따라 단어의 문법적 기능이 정해짐

교착어

- 어순이 중요시되는 영어/중국어와 달리 어근에 접사가 붙어 의미와 문법적 기능이 부여
- 굴절어의 경우에는 형태 자체가 변함으로써, 어근과 접사가 분명하게 구분되는 교착어와 다름

쉬운 예시로 우리말 "잡히시었겠더라"를 생각해 보자. 위 각주에서 설명하였듯 낱말 형성(조어)의 측면에서는 어근-접사로 나뉘고, 활용의 측면에서는 어간-어미로 나뉜다.
[2]

어근	접사				
	파생 접사	굴절 접사			
잡-	-하-	-으)시-	-었-	-겠-	-더라 ^[3]
어간	선어말 어미		어말 어미		
	어미				

각 파생+굴절 접사의 기능이 앞에서부터 피동, 주체 높임, 과거 시제, 추측, 전달임을 알 수 있다. 각각의 쓰임새가 분명하기에 여러 접사가 줄줄이 붙는다.

교착어

번호	문장	정상여부
1.	나는 밥을 먹으려 간다.	O
2.	간다 나는 밥을 먹으려.	O
3.	먹으려 간다 나는 밥을.	O
4.	밥을 먹으려 간다 나는.	O
5.	나는 먹으려 간다 밥을.	O
6.	나는 간다 밥을 먹으려.	O
7.	간다 밥을 먹으려 나는.	O
8.	간다 먹으려 나는 밥을.	O
9.	먹으려 나는 밥을 간다.	X
10.	먹으려 밥을 간다 나는.	X
11.	밥을 간다 나는 먹으려.	X
12.	밥을 나는 먹으려 간다.	O
13.	나는 밥을 간다 먹으려.	X
14.	간다 나는 먹으려 밥을.	O
15.	먹으려 간다 밥을 나는.	O
16.	밥을 먹으려 나는 간다.	O

교착어

- 접사가 붙어 같은 단어가 다양하게 생겨나기 때문에, 하나의 어근에서 생겨난 비슷한 의미의 단어가 많이 생성됨
- 따라서 이들을 모두 다르게 처리할 수 없기 때문에, 추가적인 segmentation을 통해서 같은 어근에서 생겨난 단어를 처리
- 읽을거리:
 - <http://zomzom.tistory.com/1074>
 - <https://m.blog.naver.com/reading0365/221057575669>

띄어쓰기

- 동양권에서는 띄어쓰기라는 것이 존재하지 않았고 근대에 들어 와서 도입된 것
- 띄어쓰기에 맞춰 발전 해 온 언어가 아님
- 따라서 띄어쓰기에 대한 표준이 계속 바뀌어 왔음
- 사람마다 띄어쓰기를 하는 것이 다를 뿐더러, 심지어는 띄어쓰기가 아예 없더라도 해석이 가능하기도
- 결국, 마찬가지로 추가적인 segmentation을 통해서 띄어쓰기를 정제(normalization) 해 주는 process가 필요

평서문과 의문문의 차이

언어	평서문	의문문
영어	I ate my lunch.	Did you have lunch?
한국어	점심 먹었어.	점심 먹었어?

주어 생략

언어	평서문	의문문
영어	I ate my lunch.	Did you have lunch?
한국어	점심 먹었어.	점심 먹었어?

한자 기반의 언어

언어	단어	조합
영어	Concentrate	con(=together) + centr(=center) + ate(=make)
한국어	집중(集中)	集(모을 집) + 中(가운데 중)

한자 기반의 언어

Type	Text
원문	저는 여기 한 가지 문제점이 있다고 생각합니다.
형태소에 따른 segmentation	저 는 여 기 한 가 지 문 제 점 이 있 다 고 생 각 합 니 다 .
count based subword segmentation	_저 _는 _여 기 _한 _가 지 _문 _제 _점 _이 _있 _다 고 _생 _각 _합 _니 _다 _.

한자 기반의 언어

- 문제점(問題點)
 - 문(問, 물을 문) + 제(題, 제목 제) + 점(點, 점 점)
- 결제(決濟)
 - 제(濟, 건널 제)
- 제공(提供)
 - 제(提, 끌 제)
- 제라는 token은 결국 embedding vector로 변환
 - embedding vector는 제(題, 제목 제), 제(濟, 건널 제), 제(提, 끌 제) 세 가지 모두에 대해서 embedding
 - 중앙 방향으로 vector가 애매하게 embedding 될 것

Index

- Basics
 - Probability
 - Expectation and Sampling
 - Machine Learning
 - Information
 - Gradient Based Optimization
- Natural Language Processing
 - What is NLP?
 - History of Deep Learning
 - Why NLP is Hard?
 - Why Korean NLP is Hell?
 - Recent Trends of NLP

RNNLM

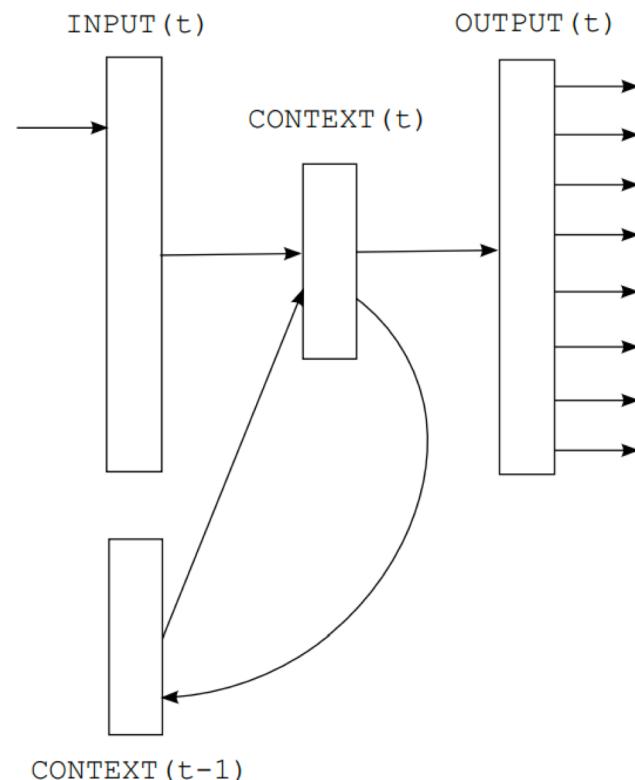
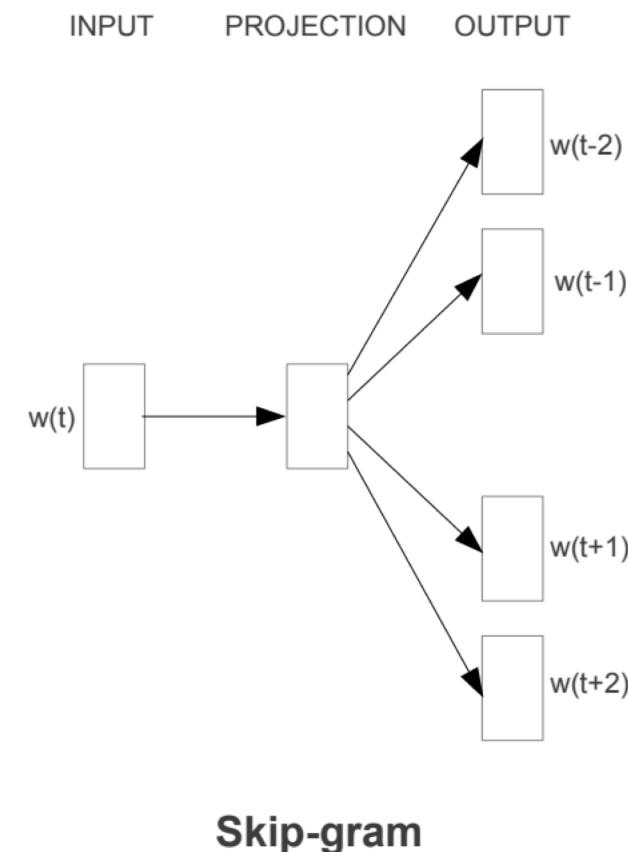
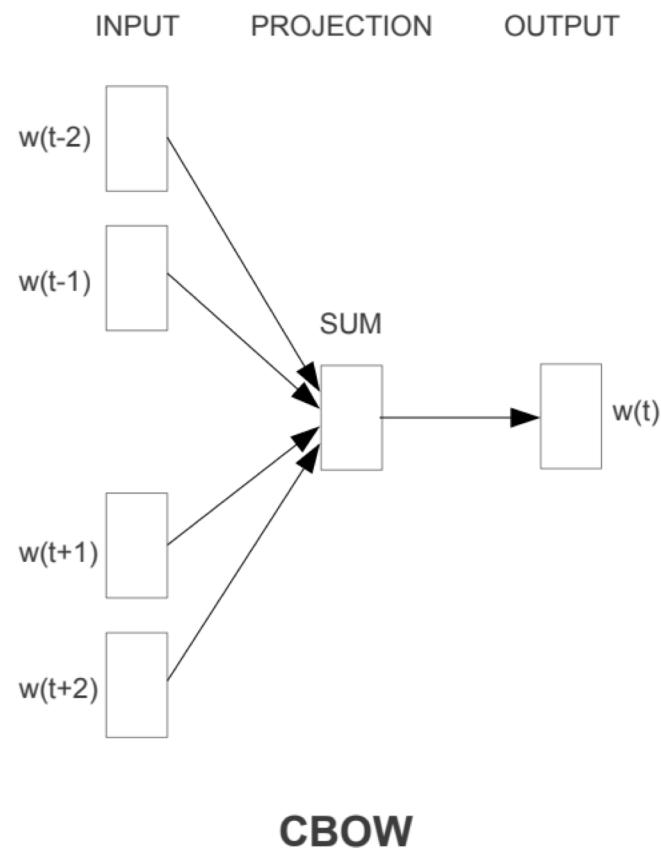


Figure 1: *Simple recurrent neural network.*

RNNLM

- 2010년에는 RNN을 활용하여 language modeling을 시도 [Mikolov et al.2010][Sundermeyer at el.2012]
 - 기존의 n-gram 기반의 language model의 한계를 극복하려 함
 - n-gram 방식과의 interpolation을 통해서 더 나은 성능의 language model을 만들어낼 수 있었지만,
 - 음성인식과 기계번역에 적용되기에에는 구조적인 한계(Weighted Finite State Transducer, WFST의 사용)로 인해서 더 큰 성과를 거둘 수는 없었음

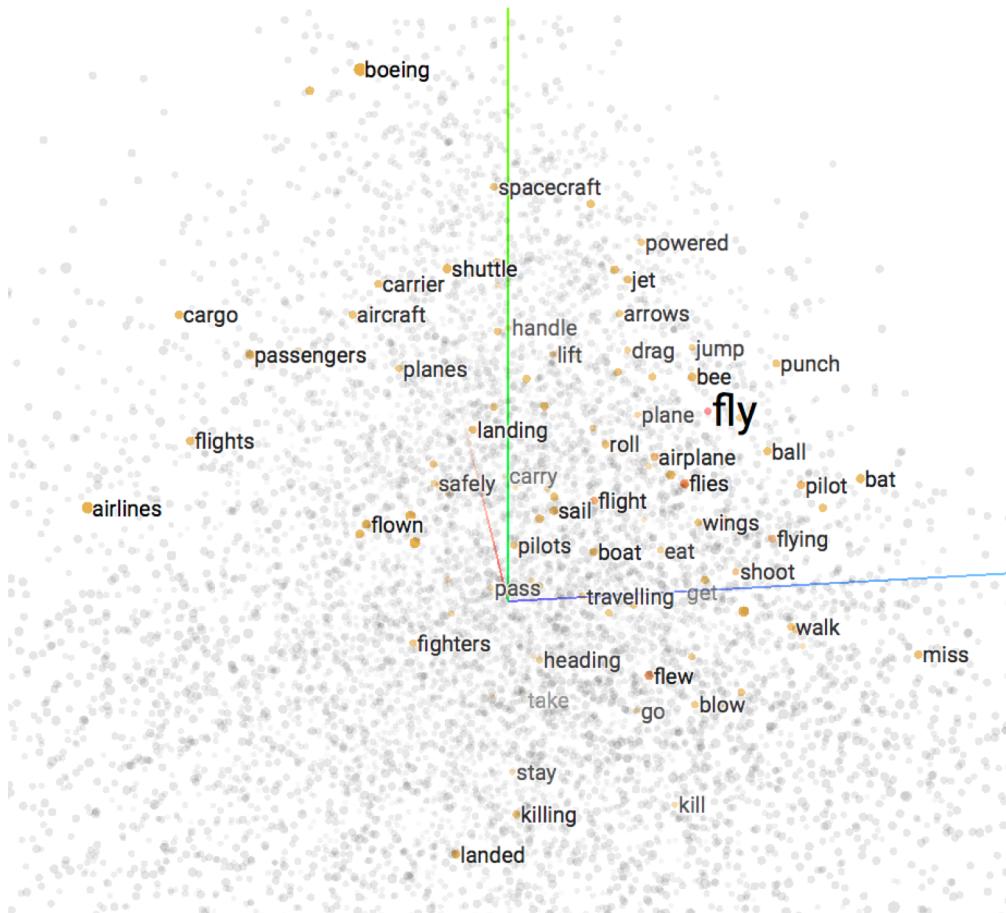
Word2Vec



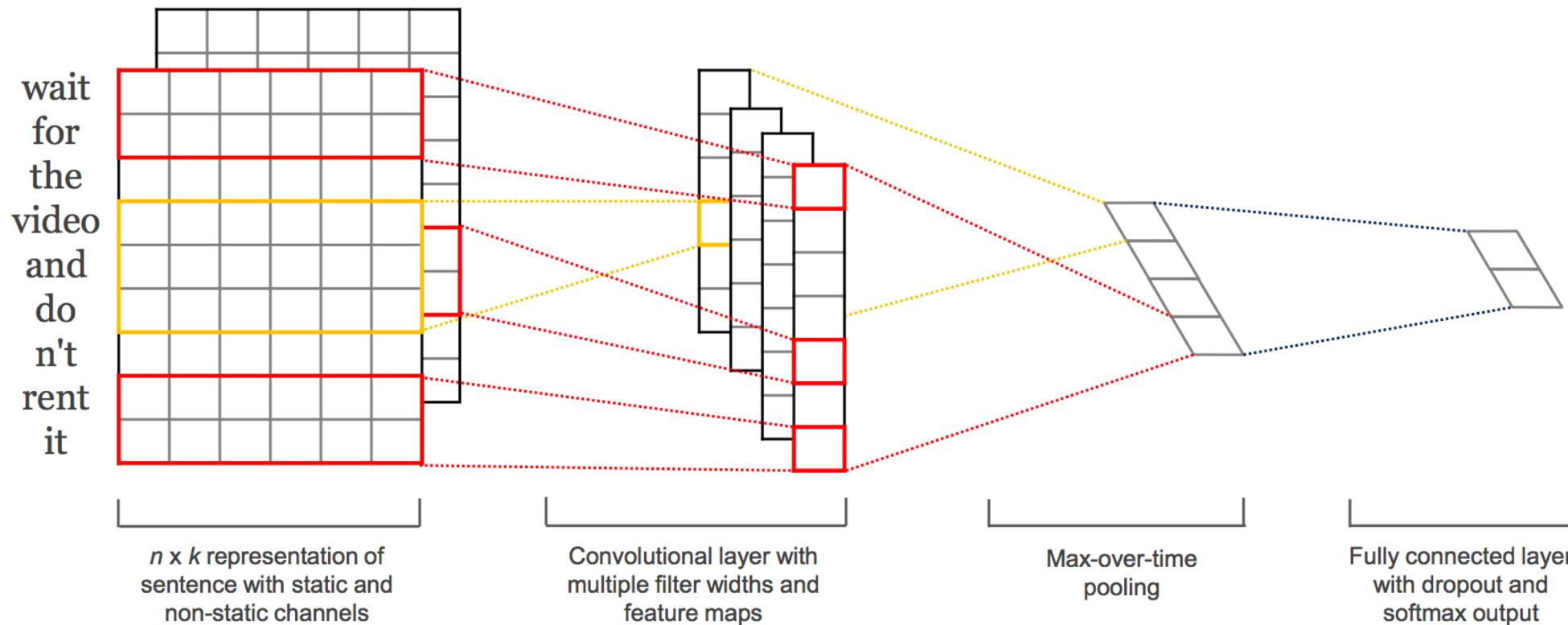
Word2Vec

- Mikolov는 2013년 Word2Vec[Mikolov et al.2013]을 발표
- 단순한 구조의 neural network를 사용하여 효과적으로 단어들을 hyper plane(또는 vector space)에 성공적으로 projection(투영)
- 본격적인 NLP 문제에 대한 딥러닝 활용의 신호탄
- 고차원의 공간에 단어가 어떻게 배치되는지 알 수 있음
- deep learning을 활용하여 NLP에 대한 문제를 해결하고자 할 때에 network 내부는 어떤 식으로 동작하는지에 대한 insight

Word2Vec



CNN on NLP



CNN on NLP

- 문장이란 단어들의 time series이기 때문에, 당연히 Recurrent Neural Network(RNN)을 통해 해결해야 한다는 **고정관념**
- 2014년, Kim은 CNN만을 활용해 기존의 Text Classification보다 성능을 끌어올린 방법을 제시[Kim et al.2014]
- word embedding vector와 결합하여 더 성능을 극대화
- NLP에 대한 시각을 한차례 더 넓힐 수 있었음

Sequence-to-Sequence

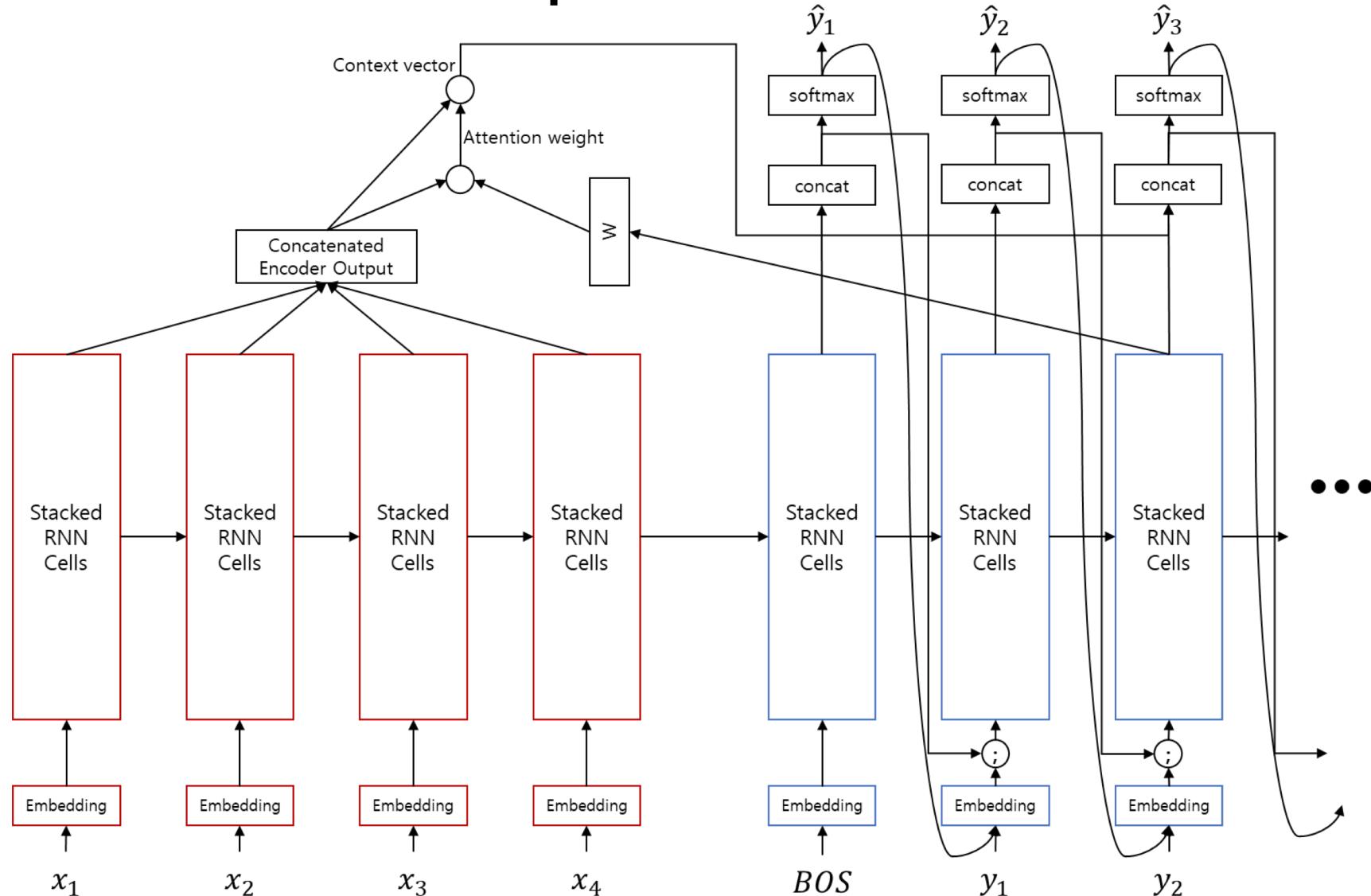


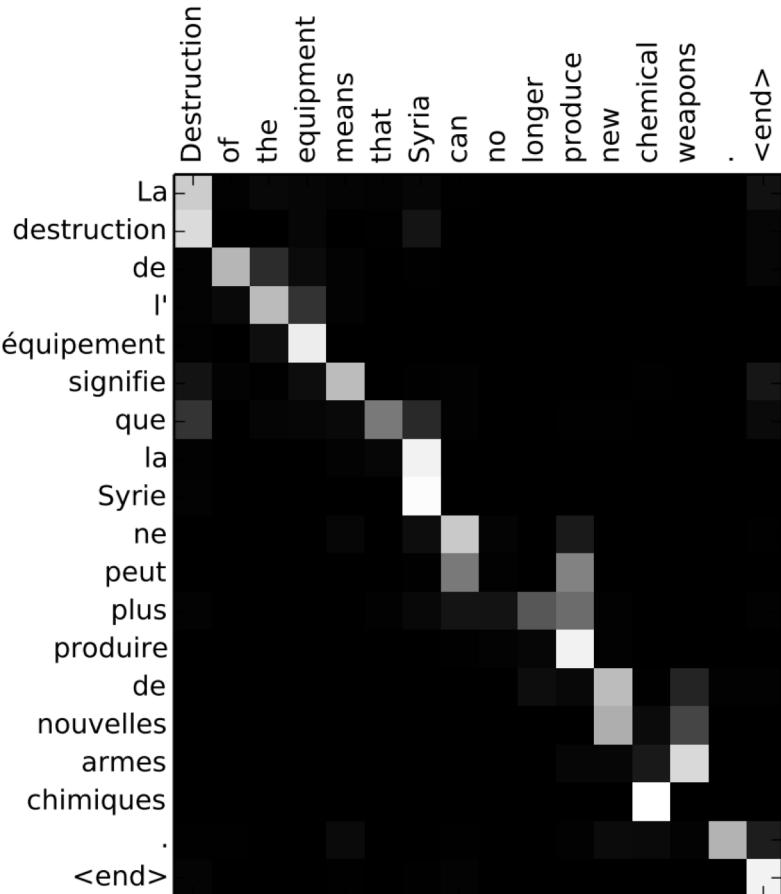
Image from GitBook

Attention

La destruction de l'équipement signifie que la Syrie ne peut plus produire de nouvelles armes chimiques.

Destruction of the equipment means that Syria can no longer produce new chemical weapons.

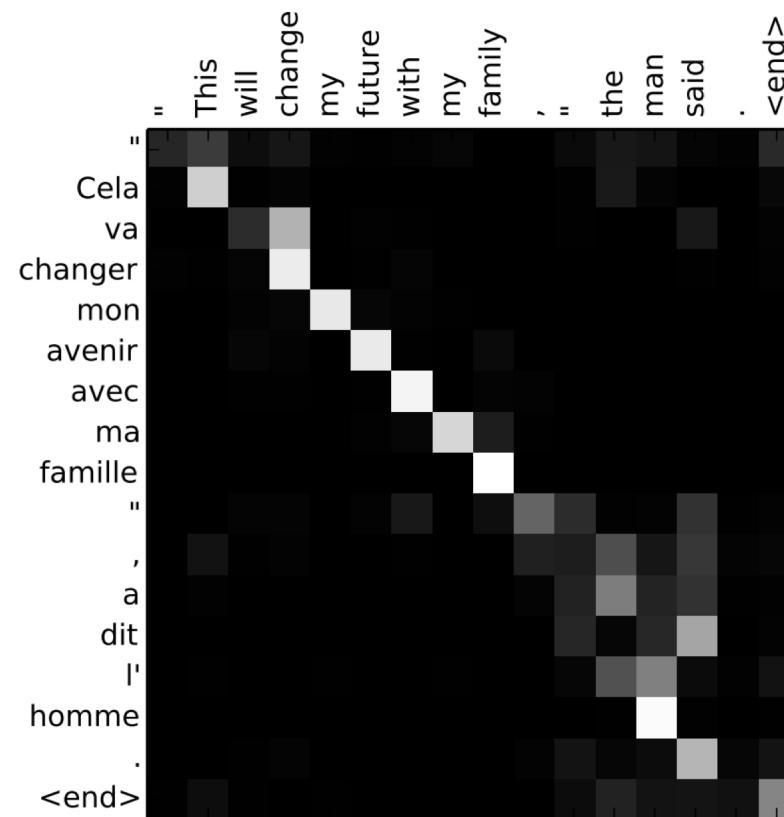
<end>

An attention visualization for a French text about Syria's chemical weapons production. The image is a 2D grid where darker shades indicate higher attention weights. The text is split into two columns: the left column contains French words, and the right column contains their English equivalents. The attention pattern highlights the correspondence between 'destruction' and 'Syria', 'équipement' and 'chemical weapons', and 'produire' and 'produce'. The grid is mostly black, with white and grey pixels concentrated along the diagonal line of word pairs.

Cela va changer mon avenir avec ma famille.

"This will change my future with my family," the man said.

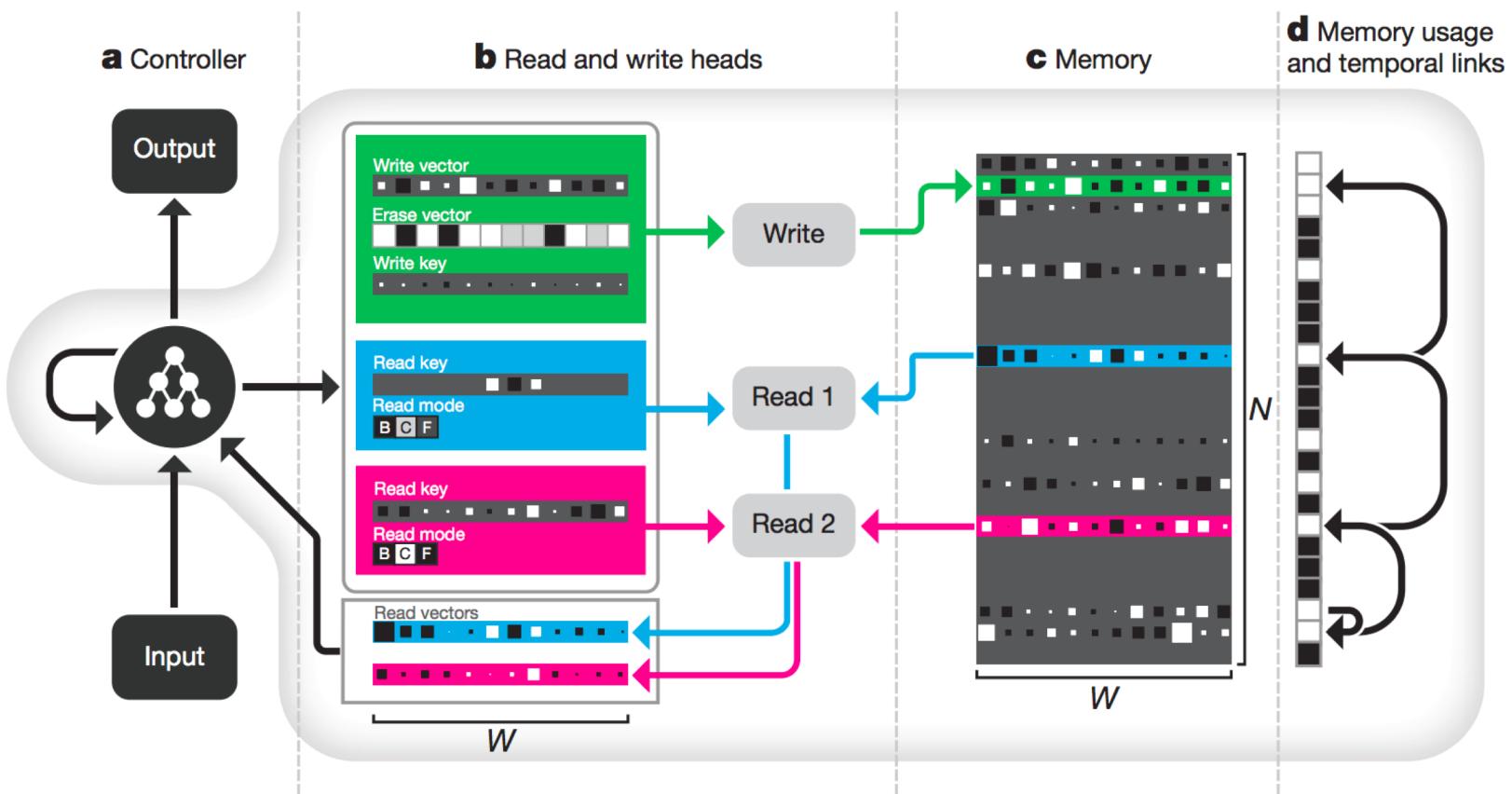
<end>

An attention visualization for an English text about a man's future. The image is a 2D grid showing attention weights between French words on the left and English words on the right. The text discusses how something will change the man's future with his family. The attention pattern shows high weights for words like 'va' (will), 'changer' (change), 'mon' (my), 'avenir' (future), 'avec' (with), 'ma' (my), 'famille' (family), and 'dit' (said). The grid is mostly black, with white and grey pixels forming a dense cluster around the central text area.

Attention

- Sequence-to-Sequence의 발표[Sutskever et al.2014]에 이어, Attention 기법이 개발되어 성공적으로 기계번역에 적용[Bahdanau et al.2014]하여 큰 성과
- 기존의 한정적인 적용 사례에서 벗어나, 주어진 정보에 기반하여 자유롭게 문장을 생성할 수 있게 된 것
- 기계번역 뿐만 아니라, summarization, 챗봇 등 더 넓고 깊은 주제의 NLP의 문제를 적극적으로 해결해보려 시도 할 수 있게 됨
- 더욱더 많은 연구가 활기를 띠게 되어 관련한 연구가 쏟아져 나옴
- 기계번역은 가장 먼저 end-to-end 방식을 활용하여 상용화에 성공하였을 뿐만 아니라, NLP에 대한 이해도가 더욱 높아지게 됨

Memory Augmented Neural Network (MANN)



Memory Augmented Neural Network (MANN)

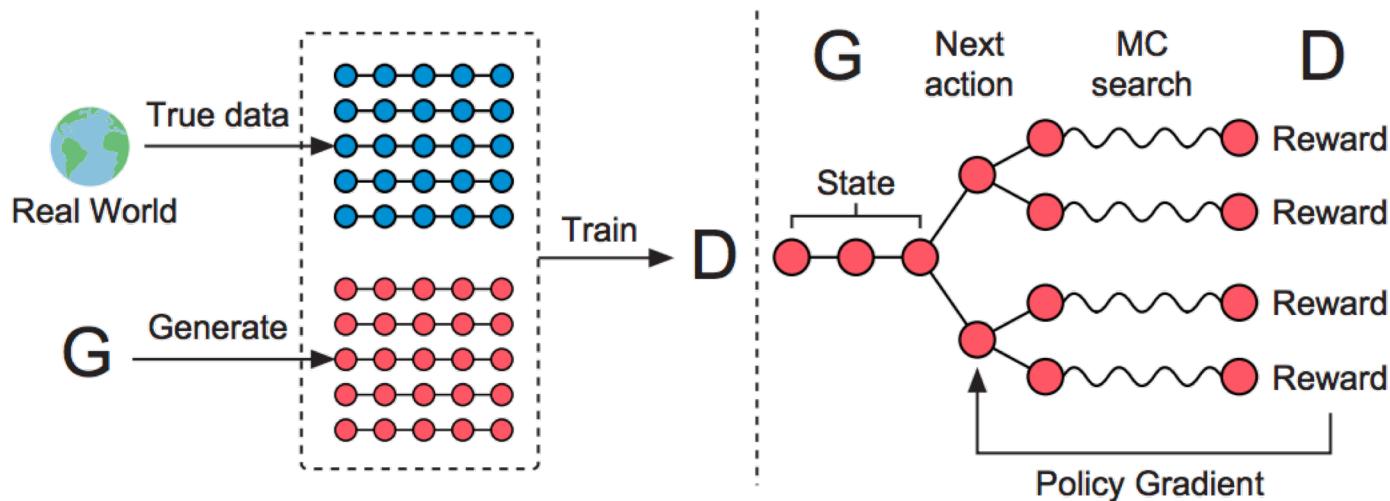
- Attention이 큰 성공을 거두자, continuous한 방식으로 memory에 access하는 기법에 대한 관심이 커짐
- Neural Turing Machine(NTM)[Graves et al.2014]
 - Continuous한 방식으로 memory에서 정보를 read/write하는 방법을 제시
- Differential Neural Computer (DNC)[Graves et al.2016]
- 읽을거리:
 - <https://jamiekang.github.io/2017/05/08/neural-turing-machine>
 - <https://sites.google.com/view/mann-emnlp2017/>

NLP and Reinforcement Learning

- Generative Learning on Computer Vision
 - Variational Auto Encoder(VAE)[Kingma et al.2013]
 - Generative Adversarial Networks(GAN)[Goodfellow et al.2014]
 - 기존의 discriminative learning 방식을 벗어나 generative learning에 관심
- NLP분야는 그럴 필요가 없었음
 - Language modeling 자체가 문장에 대한 generative learning

NLP and Reinforcement Learning

- BUT 기계번역은 어려움에 부딪히게 됨
 - Deep learning에서 사용하는 training objective와 실제 기계번역을 위한 objective function과 괴리(discrepancy)가 있었기 때문
 - 따라서, 마치 Computer Vision에서 **기존의 MSE loss의 한계를 벗어나기 위해 GAN을 도입한 것처럼**, 기존의 loss function과 다른 무엇인가가 필요



NLP and Reinforcement Learning

- 성공적으로 강화학습의 **policy gradients** 방식을 NLP에 적용
[Bahdanau et al.2016][Yu et al.2016]
- 강화학습(RL)을 사용하여 실제 task에서의 objective function으로부터 **reward를 받을 수 있게 됨**에 따라, 더욱 성능을 극대화
- Adversarial learning을 NLP에서도 수행 가능

So, what is NLG?

- NLP에서 가장 강력한 분야
 - 응용 될 여지가 가장 많음
 - 실제로 지금도 다양한 분야에 널리 적용
 - 음성인식, OCR, 기계번역, QnA
 - 아직 나아갈 길이 멀다
 - 사람을 돋고 **편리하게 하기 위한** 과정
 - NLU도 중요하지만 결국 중요한 것은 **문장이 출력** 되는 것
 - End-to-end 모델을 추구하므로, NLU도 하나의 모델로 합쳐지지 않을까?
 - 결국 **컴퓨터와 사람 사이의 인터페이스**로, 사람이 사용하는 언어를 만 들어낼 수 있어야 한다