

Natural Language Processing with Deeplearning



Figure 1: TARS from Interstellar – Image from web

What is Natural Language Processing?

자연어처리(natural language processing, NLP) 분야는 인공지능의 큰 줄기 중에 하나입니다. 특히, 컴퓨터에게 사람이 사용하는 언어를 처리하고 이해하도록 함으로써, 사람과 컴퓨터 사이의 매개체 또는 인터페이스 역할을 할 수 있습니다. 따라서 computer science 뿐만 아니라, linguistics와 같은 다른 학문과의 융합적인 요소도 갖고 있습니다. 실제로 NLP는 아래와 같이 세부적인 주제로 나누어 볼 수 있습니다.

- Phonetics and Phonology – the study of linguistic sounds

- Morphology – the study of the meaning of components of words
- Syntax – the study of the structural relationships between words
- Semantics – the study of meaning
- Discourse – they study of linguistic units larger than a single utterance

[Gao et al.2017]

따라서, 이러한 NLP의 세부적인 부분들이 합쳐져 최종적인 목표는 사람의 언어를 이해하여 컴퓨터로 하여금 여러가지 tasks를 수행할 수 있도록 하는 것입니다. 컴퓨터는 이제 우리와 뗄 수 없는 존재가 되었고, 그러므로 이미 실제로 NLP는 우리의 일상에 가장 깊숙히 들어와 있는 분야이기도 합니다. NLP에 의해서 수행되는 대표적인 task 또는 응용분야들은 다음과 같습니다.

- 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

우리는 이 책을 통해서 위의 task들을 위한 기술들의 대부분을 다루고자 합니다. 위의 대부분의 기술들이 deep learning에 의해서 비약적인 발전이 있었지만, 그 기반이 되는 수십년의 역사 또한 중요합니다. 따라서 최신 기술에 대해서 다루기 위해서는 그 이전의 기술들에 대해서도 다루고, 무엇이 문제였으며, 어떻게 최신의 기술이 어떤 돌파구를 마련했는지 아는 것도 중요합니다. 따라서, 이 책은 이러한 task들에 대한 최신 기술 뿐만 아니라, deep learning 이전의 주요 기술들에 대해서도 간략히 다루어 기초부터 차근차근 쌓아올릴 수 있도록 하고자 합니다. # Deep Learning

딥러닝의 시대가 오고 딥러닝은 하나하나 머신러닝의 분야들을 정복해 나가기 시작했습니다. 가장 먼저 두각을 나타낸 곳은 ImageNet이었지만, 가장 먼저 상용화 부문에서 빛을 본 것은 음성인식 분야였습니다. 음성인식은 여러 components 중에서 고작 하나인 GMM을 DNN으로 대체하였지만, 성능에 있어서 십수년의 정체를 뚫고 한 차례 큰 발전을 이루어냈습니다. 상대적으로 가장 나중에 빛을 본 곳은 NLP분야였습니다. 아마도 image classification과 음성인식의 phone recognition과 달리 NLP는 sequential한 데이터라는 것이 좀 더 장벽으로 다가왔으리라 생각됩니다. 하지만, 결국엔 attention의 등장으로 인해서 요원해 보이던 기계번역 분야마저 end-to-end deep learning에 의해서 정복되게 되었습니다.

Brief Introduction to History of Deep Learning

Before 2010's

인공신경망을 위시한 인공지능의 유행은 지금이 처음이 아닙니다. 이전까지 두 번의 대유행이 있었고, 그에 따른 두 번의 빙하기가 있었습니다. 80년대에 처음 back-propagation이 제안된 이후로, 모든 문제는 해결 된 듯 해 보였습니다. 하지만, 다시금 여러가지 한계점을 드러내며 침체기를 맞이하였습니다. 모두가 인공신경망의 가능성을 부인하던 2006년, Hinton 교수는 Deep Belief Networks을 통해 여러 층의 hidden layer를 효과적으로 pretraining 시킬 수 있는 방법을 제시하였습니다. 하지만, 아직까지 가시적인 성과가 나오지 않았기 때문에 모두의 관심을 집중 시킬 순 없었습니다. 아~ 그런가보다 하고 넘어가는 수준이었겠지요.

실제로 주변의 90년대의 빙하기를 겪어보신 세대 분들은 처음 딥러닝이 주목을 끌기 시작했을 때, 모두 부정적인 반응을 보이기 마련이었습니다. 계속 해서 최고 성능을 갈아치우며, 모두가 열광할 때에도, 단순한 잠깐의 유행일 것이라 생각하는 분들도 많았습니다. 하지만 점차 딥러닝은 여러 영역을 하나둘 정복 해 나가기 시작했습니다.

Image Recognition

2012년 이미지넷에서 인공신경망을 이용한 AlexNet([Krizhevsky et al. 2012])은 경쟁자들을 큰 차이로 따돌리며 우승을 하고, 딥러닝의 시대의 서막을 올립니다. AlexNet은 여러 층의 Convolutional Layer을 쌓아서 architecture를 만들었고, 기존의 우승자들과 확연한 실력차를 보여주었습니다. 당시에 AlexNet은 3GB 메모리의 Nvidia GTX580을 2개 사용하여 훈련하였는데, 지금 생각하면 참으로 격세지감이 아닐 수 없습니다.

이후, ImageNet은 딥러닝의 경연장이 되었고, 거의 모든 참가자들이 딥러닝을 이용하여 알고리즘을 구현하였습니다. 결국, ResNet([He et al. 2015])은 Residual Connection을 활용하여 150층이 넘는 deep architecture를 구성하며 우승하였습니다.

하지만, 사실 연구에서와 달리, 아직 실생활에서의 image recognition은 아직 다른 분야에 비해서 어려움이 있는 것은 사실입니다. image recognition 자체의 어려움이 워낙 높기 때문입니다. 따라서 아직도 이와 관련해서 산업계에서는 많은 연구와 개발이 이어지고 있습니다.

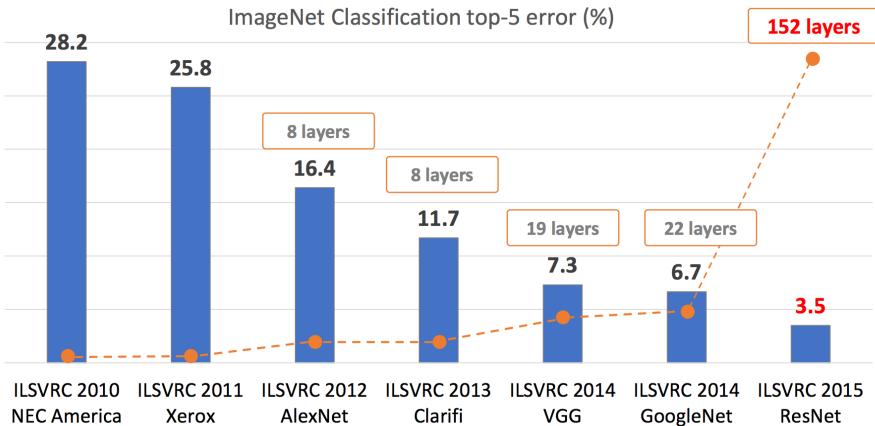


Figure 2: Recent History of ImageNet

Speech Recognition

음성인식에 있어서도 딥러닝(당시에는 Deep Neural Network라는 이름으로 더욱 유명하였습니다.)을 활용하여 큰 발전을 이룩하였습니다. 오히려 이 분야에서는 vision분야에 비해서 딥러닝 기술을 활용하여 상용화에까지 성공한 더욱 인상적인 사례라고 할 수 있습니다.

사실 음성인식은 2000년대에 들어 큰 정체기를 맞이하고 있었습니다. GMM(Gaussian Mixture Model)을 통해 phone을 인식하고, 이를 HMM(Hidden Markov Model)을 통해 sequential하게 modeling하여 만든 Acoustic Model (AM)과 n-gram기반의 Language Model (LM)을 WFST(Weighted Finite State Transducer)방식을 통해 결합하는 전통적인 음성인식(Automatic Speech Recognition, ASR) 시스템은 위의 설명에서 볼 수 있듯이 너무나도 복잡한 구조와 함께 그 성능의 한계를 보이고 있었습니다.

그러던 중, 2012년 GMM을 DNN으로 대체하며, 십수년간의 정체를 단숨에 뛰어넘는 큰 혁명을 맞이하게 됩니다. (Vision, NLP에서 모두 보이는 익숙한 패턴입니다.) 그리고 점차 AM전체를 LSTM으로 대체하고, 또한 end-to-end model([Chiu et al.2017])이 점점 저변을 넓혀가고 있는 추세입니다.

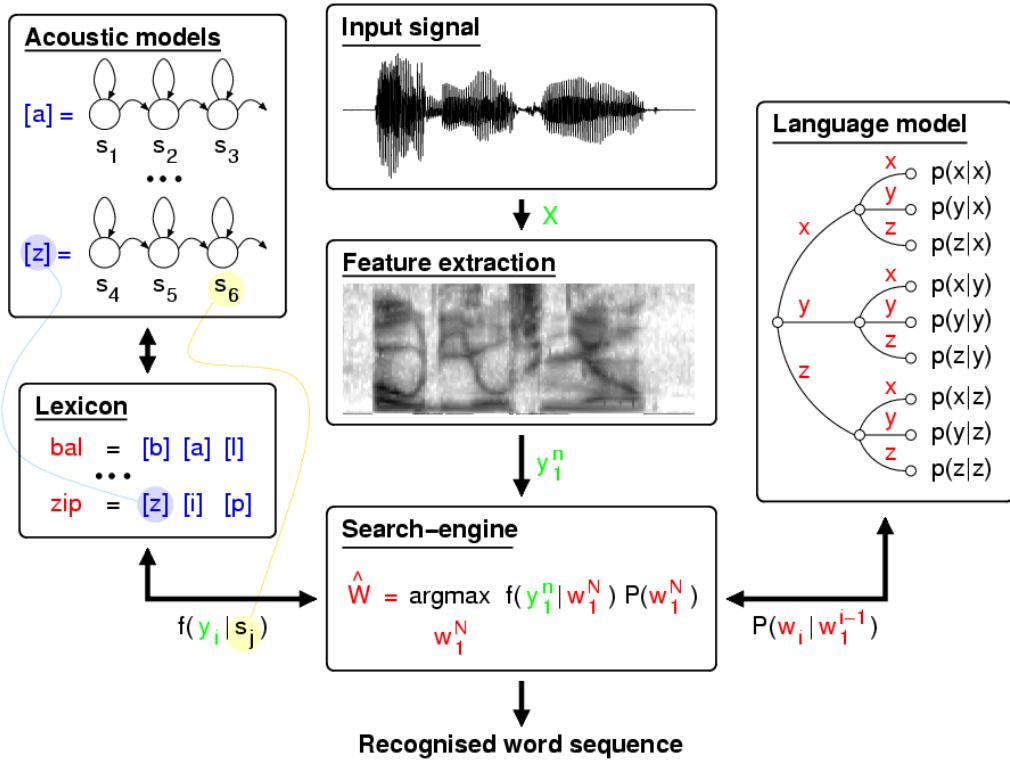


Figure 3: Traditional Speech Recognition System

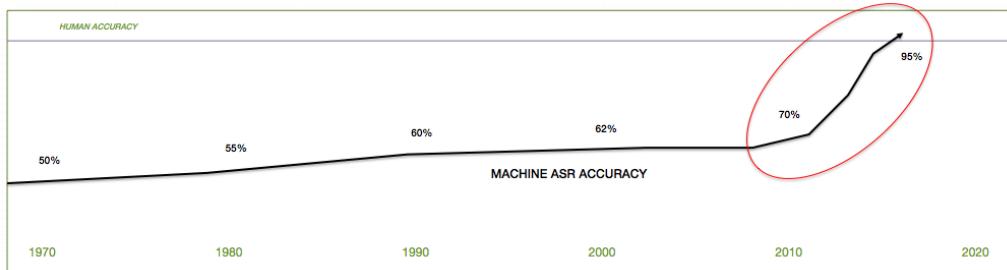


Figure 4: Accuracy of ASR



Figure 5: It was hard time for NLP until 2014.

Machine Translation

사실 다른 인공지능의 다른 분야에 비해서 NLP 또는 기계번역 분야는 이렇다할 큰 성과를 거두지는 못하고 있었습니다. 하지만 결국 물밀듯이 밀려오는 딥러닝의 침략 앞에서 기계번역 또한 예외일 순 없었습니다. 딥러닝 이전의 기계번역은 통계 기반 기계번역(Statistical Machine Translation, SMT)가 지배하고 있었습니다. 비록 SMT는 규칙기반의 번역방식(Rule based Machine Translation, RBMT)에 비해서 언어간 확장이 용이한 장점이 있었고, 성능도 더 뛰어났지만, 음성인식과 마찬가지로 SMT는 역시 너무나도 복잡한 구조를 지니고 있었습니다.

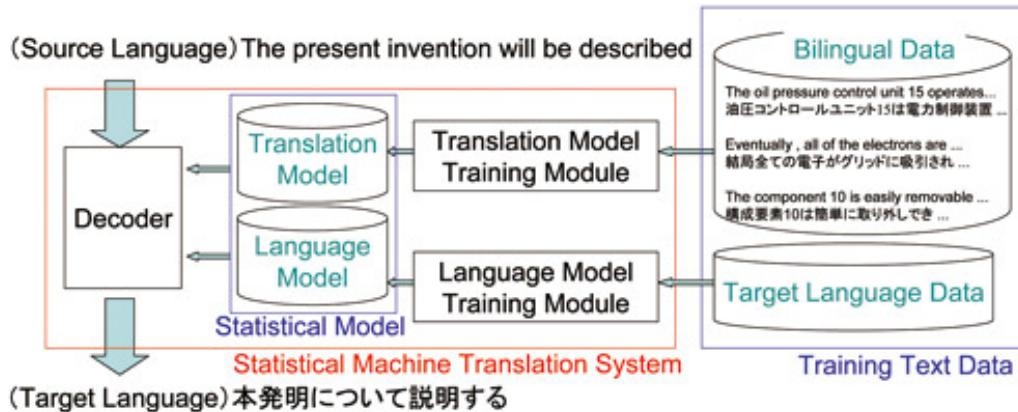


Figure 6: Sub-modules for Statistical Machine Translation (SMT)

2014년 Sequence-to-sequence(seq2seq)라는 architecture가 소개 되며, end-to-end neural machine translation의 시대가 열리게 되었습니다.

Seq2seq를 기반으로 attention mechanism([Bahdanau et al.2014], [Luong et al.2015])이 제안되며 결국 기계번역은 Neural Machine Translation에 의해서 대통합이 이루어지게 됩니다.

결국, 기계번역은 가장 늦게 혁명이 이루어졌지만, 가장 먼저 딥러닝만을 사용해 상용화가 된 분야가 되었습니다. 현재의 상용 기계번역 시스템은 모두 딥러닝에 의한 시스템으로 대체되었다고 볼 수 있습니다.

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/>

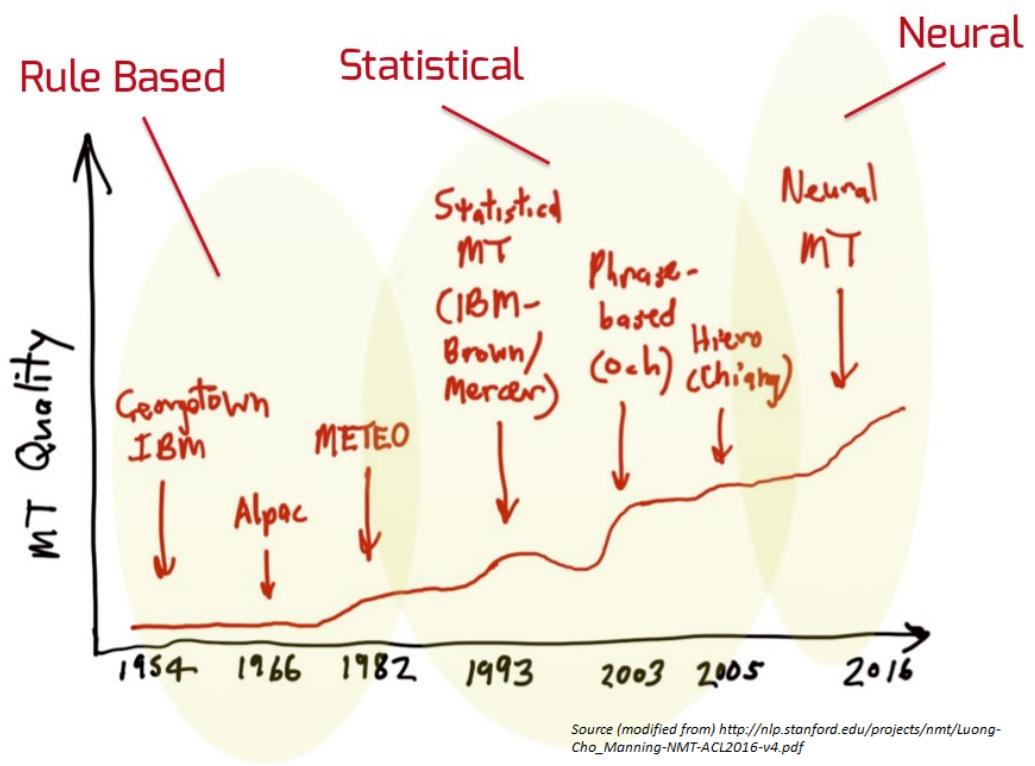


Figure 7: History of Machine Translation

Generative Learning

Neural Network은 pattern classification에 있어서 타 알고리즘에 비해서 너무나도 암도적인 성능을 보여주었기 때문에, image recognition, text classification과 같은 단순한 분류 문제(classification or discriminative learning)는 금방 정복되고 더 이상 연구자들의 흥미를 끌 수 없었습니다.

각 방식이 흥미를 두고 있는 것:

- Discriminative learning

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

- Generative learning

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

따라서, 곧 연구자들은 또 다른 흥미거리를 찾아 나섰는데, 그것은 Generative Learning이었습니다. 기존의 classification 문제는 X 가 주어졌을 때, 알맞은 Y 를 찾아내는 것에 집중했다면, 이제는 X 자체에 집중하기 시작한 것입니다. 예를 들어 기존에는 사람의 얼굴 사진이 주어지면 남자인지 여자인지, 또는 더 나아가 이 사람이 누구인지 알아내는 것이었다면, 이제는 얼굴 자체를 묘사할 수 있는 모델을 훈련하고자 하였습니다.

이러한 과정에서 Adversarial learning (GAN, [Goodfellow et al.2014])이나 Variational Auto-encoder (VAE, [Kingma et al.2013])등이 주목받게 되었습니다. 아직 이러한 연구는 현재 진행형이라 할 수 있고, 이와 관련한 많은 문제들이 남아있습니다.

Paradigm Shift on NLP from Traditional to Deep Learning

Deep learning 이전의 기존의 전형적인 NLP application의 구조는 보통 아래와 같습니다. Task에 따라서 phonology가 추가되기도 하고, 아래와 같이 여러가지 단계의 module로 구성되어 복잡한 디자인을 구성하게 됩니다. 따라서 매우 무겁고 복잡하여 구현 및 시스템 구성이 어려운 단점이 많았습니다. 더군다나, 각각의 module이 완벽하게 동작할 수 없기 때문에, 각기 발생한 error가 중첩 및 가중되어 뒤로 전파되는 error propagation등의 문제도 가질 수 있었습니다.



Figure 8: generated by a progressively grown GAN trained on the CelebA–HQ dataset

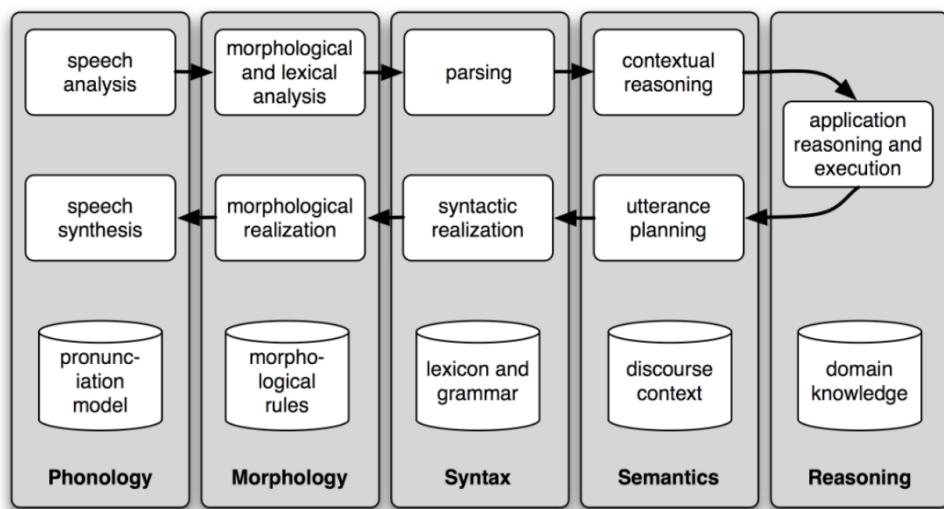


Figure 9: [Gao et al.2017]

하지만, 위에서 언급한 기계번역의 사례처럼 NLP 전반에 걸쳐 deep learning의 물결이 들어오기 시작했습니다. 처음에는 각 sub-module을 대체하는 형태로 진행되었지만, 점차 기계번역의 사례처럼 결국 end-to-end model들로 대체되었습니다. 현재에도 chat-bot과 같은 아직 많은 task들에서 end-to-end learning이 이루어지지 않았지만, 최종적으로는 end-to-end model이 제안될 것이라 볼 수 있습니다.

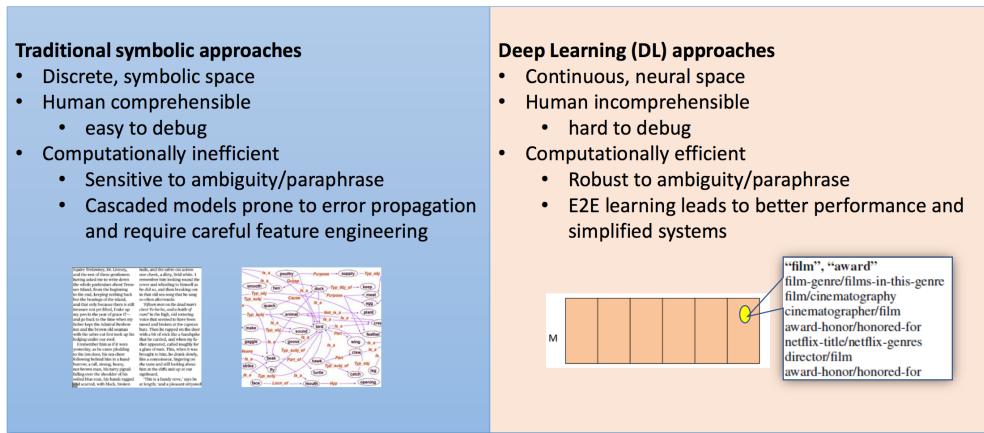


Figure 10: [Gao et al.2017]

Deep learning이 NLP에서도 주류가 되면서, 위와 같은 접근 방법의 변화들을 꼽을 수 있습니다. 사람의 언어는 Discrete한 symbol로 이루어져 있습니다. 비록 그 symbol간에는 유사성이 있을 수 있지만 기본적으로 모든 단어(또는 token)은 다른 symbol이라고 볼 수 있습니다. 따라서 기존의 전통적인 NLP에서는 discrete symbol로써 데이터를 취급하였습니다. 따라서 사람이 데이터를 보고 해석하기는 쉬운 장점이 있었지만, 모호성이나 유의성을 다루는데에는 어려움을 겪을 수 밖에 없었습니다.

하지만 word2vec등의 word embedding을 통해서 단어(또는 token)을 continuous한 vector로써 나타낼 수 있게 되고, 모호성과 유의성에서도 이득을 볼 수 있게 되었습니다. 또한, deep learning의 장점을 잘 살려 end-to-end model을 구현함으로써 더욱 높은 성능을 뽑을 수 있게 되었습니다. 또한, RNN의 단점을 보완한 LSTM과 GRU에 대한 활용법이 고도화 되었고, attention의 등장으로 인해서 긴 time-step의 sequential 데이터에 대해서도 어렵지 않게 훈련할 수 있게 된 점도 큰 터닝 포인트라고 볼 수 있습니다.

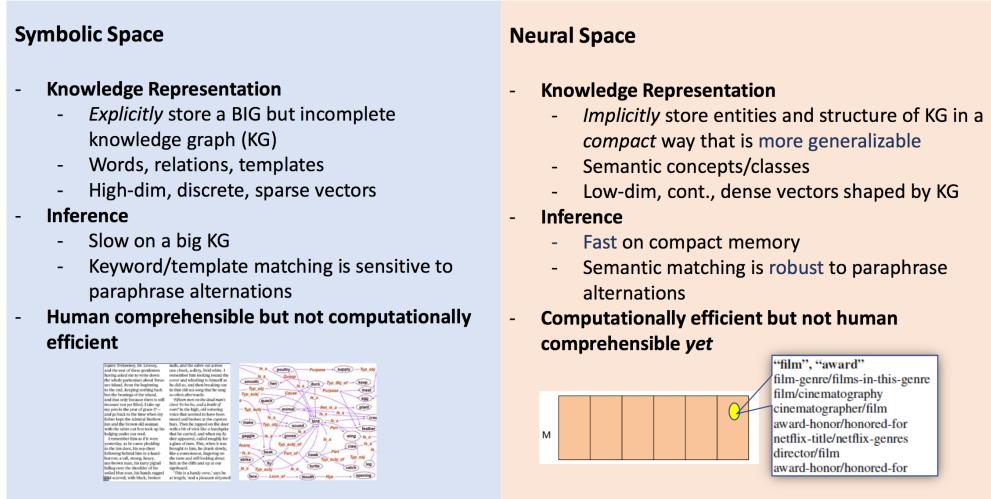


Figure 11: [Gao et al.2017]

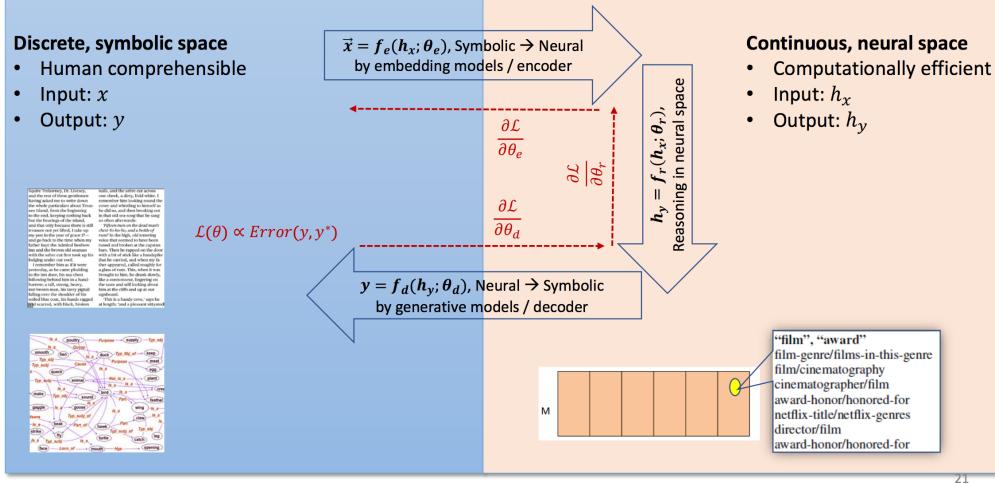
Conclusion

다시 말해, 비록 NLP는 discrete한 symbol로 이루어진 사람의 언어를 다루는 분야이지만, 성공적인 word embedding과 long term sequential data에 대한 효과적인 대응방법이 나옴에 따라서, 점차 다른 인공지능의 분야들처럼 큰 발전이 이루어지게 되었습니다.

이렇게 deep learning이 널리 퍼짐에 따라, NLP를 포함한 인공지능의 여러 분야에서 모두 큰 발전과 성공을 보이고 있습니다. 따라서, 이 책은 기존의 전통적인 방식과 새롭게 제안된 최신의 deep learning 기술을 모두 소개하고자 합니다.

Why NLP is difficult?

음성인식은 눈에 보이지 않는 signal을 다룹니다. 보이지도 않는 가운데 noise와 signal을 가려내야 하고, 소리의 특성상 noise와 signal은 그냥 더해져서 나타납니다. 게다가 time-step은 무지하게 길어요. 어려울 것 같습니다. 그렇다면 눈에 보이는 computer vision을 생각 해 보죠. 그런데 computer vision은 눈에 보이지만 이미지는 너무 크고, 다양합니다. 심지어 내 눈에는 다 똑같은 색깔인데 사실은 알고보면 다른 색이라고 합니다. 그럼 애초에 discrete한 단어들로 이루어져 있는 자연어처리를 해



21

Figure 12: [Gao et al.2017]

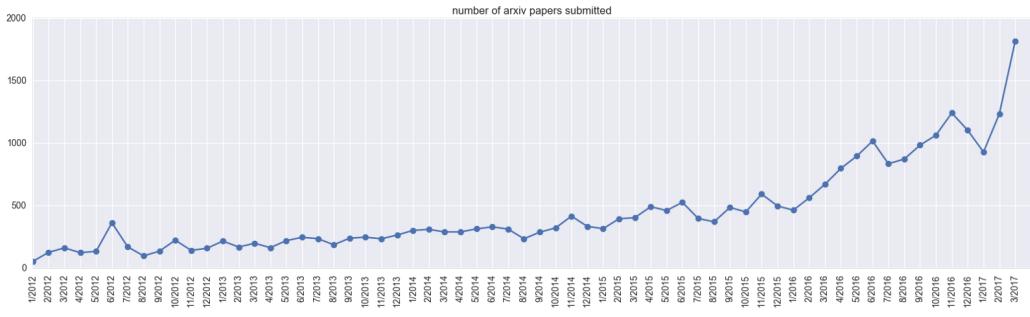


Figure 13: A Number of submitted papers to Arxiv. Image from Karpathy's medium

볼까요? 그럼 NLP는 쉬울까요? 하지만 세상에 쉬운일은 없죠… Natural language processing도 다른 분야 못지않게 매우 어렵습니다. 어떠한 점들이 NLP를 어렵게 만드는 것일까요?

사람은 언어를 통해 타인과 교류하고, 의견과 지식을 전달 합니다. 소리로 표현된 말을 석판, 나무, 종이에 적기 시작하였고 사람의 지식은 본격적으로 축적되기 시작하였습니다. 이와 같이 언어는 사람의 생각과 지식을 내포하고 있습니다. 컴퓨터로 하여금 이러한 사람의 언어를 이해할 수 있게 한다면 컴퓨터에게도 지식과 의견을 전달 할 수 있을 것 입니다.

Ambiguity

아래의 문장을 한번 살펴볼까요. 어떤 회사의 번역이 가장 정확한지 살펴 볼까요.
(2018년 4월 기준)

차를 마시러 공원에 가던 차 안에서 나는 그녀에게 차였다. – Google: I was kicking her in the car that went to the park for tea. – Microsoft: I was a car to her, in the car I had a car and went to the park. – Naver: I got dumped by her on the way to the park for tea. – Kakao: I was in the car going to the park for tea and I was in her car. – SK: I got dumped by her in the car that was going to the park for a cup of tea.

안타깝게도 완벽한 번역은 없는 것 같습니다. 같은 차라는 단어가 세 번 등장하였고, 모두 다른 의미를 지니고 있습니다: tea, car, and kick (or dump). 일부는 표현을 빼드리기도 하였고, 다른 일부는 단어를 헷갈린 것 같습니다. 이렇게 단어의 중의성 때문에 문장을 해석하는데 모호함이 생기기도 합니다. 또 다른 상황을 살펴보겠습니다.

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.

위와 같이 문장 내 정보의 부족으로 인한 모호성이 발생 할 수 있습니다. 사람의 언어는 효율성을 극대화 하도록 발전하였기 때문에, 최소한의 표현으로 최대한의 정보를 표현하려 합니다. 따라서 앞뒤 문장의 context에 따라서 문장의 의미는 달라질 것입니다. 사실 첫 예제의 차도 주변 단어들(context)를 보면 중의성을 해소(word sense disambiguation)할 수 있습니다. 아래의 예제도 문장 내 정보의 부족이 야기한 구조 해석의 문제입니다.

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

Paraphrase



Figure 14:

김치 싸대기로 유명한 드라마 모두 다 김치의 문제적 장면

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.
8. 김치가 여자에게 무기로 사용되어 남자를 후려치고 있다.
9. 김치가 여자에게서 남자에게로 날라가고 있다.

영화나 드라마의 어떤 장면을 말로 표현한다고 해 봅시다. 그럼 아주 다양한 표현이 나올 것 입니다. 하지만 알고보면 다 같은 장면을 묘사하고 있는 것이고,

그 안에 포함된 의미는 같다고 할 수 있을 것 입니다. 이와 같이 문장의 표현 형식은 다양하고, 비슷한 의미의 단어들이 존재하기 때문에 paraphrase의 문제가 존재합니다. 더군다나, 위에서 지적 한 것 처럼 미묘하게 사람들이 이해하고 있는 단어의 의미는 다를 수도 있을 것 입니다. 따라서 이 또한 더욱 paraphrase 문제의 어려움을 가중시킵니다.

Discrete, Not Continuous

사실은 discrete하기 때문에 그동안 쉽다고 느껴졌습니다. 하지만 neural network에 적용하기 위해서는 continuous한 값으로 바꾸어야 합니다. Word embedding이 그 역할 훌륭하게 수행하고 있긴 합니다. 하지만 애초에 continuous한 값이 아니었기 때문에 neural network 상에서 여러가지 방법을 구현할 때에 제약이 존재합니다.

Curse of Dimensionality

Discrete한 데이터이기 때문에 많은 종류의 데이터를 표현하기 위해서는 엄청난 dimension이 필요합니다. 예전에는 각 단어를 discrete한 symbol로 써 다루었기 때문에, 마치 vocabulary size = $|V|$ 만큼의 dimension이 있는 것이나 마찬가지였습니다. 이러한 sparseness를 해결하기 위해서 단어를 적절하게 segmentation하는 등 여러가지 노력이 필요하였습니다. 다행히 적절한 word embedding을 통해서 dimension reduction을 하여 이 문제를 해결함으로써, 이제는 이러한 문제는 예전보다 크게 다가오진 않습니다.

Noise and Normalization

모든 분야의 데이터에서 noise를 signal로 부터 적절히 분리해 내는 일은 매우 중요합니다. 그러한 과정에서 자칫 실수하면 data는 본래의 의미마저 같이 잃어버릴 수도 있습니다. 이러한 관점에서 NLP는 어려움이 존재 합니다. 특히, 다른 종류의 데이터에 비해서 데이터가 살짝 바뀌었을 때의 의미의 변화가 훨씬 크기 때문입니다. 예를 들어 이미지에서 한 픽셀의 RGB값이 각각 0에서 255까지로 나타내어지고, 그 값중 하나의 수치가 1이 바뀌었다고 해도 해당 이미지의 의미는 변화가 없다고 할 수 있습니다. 하지만 단어는 discrete한 symbol이기 때문에, 단어가 살짝만 바뀌어도 문장의 의미가 완전히 다르게 변할 수도 있습니다. 또한, 마찬가지로 띠어쓰기나 어순의 차이로 인한 정제의 이슈도 큰 어려움이 될 수 있습니다. 이러한 어려움을

다루고 해결하기 위한 방법을 Preprocessing 챕터에서 다루도록 하겠습니다. # 무엇이 한국어 NLP를 더욱 어렵게 만드는가?

교착어 (어순)

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

한국어는 교착어에 속합니다. 어순이 중요시되는 영어/중국어와 달리 어근에 접사가 붙어 의미와 문법적 기능이 부여됩니다. (굴절어의 경우에는 형태 자체가 변함으로써, 어근과 접사가 분명하게 구분되는 교착어와 다릅니다.) 따라서, 아래와 같은 재미있는 예시의 형태도 가능합니다.

쉬운 예시로 우리말 "잡하시었겠더리"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		
접-	-하-	-으(으)시-	-았-	-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.

Figure 15: [Image from 나무위키(교착어)]

위의 예처럼 접사에 따라 단어의 역할이 정의되기 때문에, 상대적으로 어순은 중요하지 않습니다. 아래는 4개의 단어가 나타날 수 있는 모든 조합을 적은 것입니다. “간다”가 “밥을”뒤에 붙어 수식할 때를 제외하면 모두 같은 의미의 문장이 됩니다.

번호	문장	정상여부
1.	나는 밥을 먹으러 간다.	O
2.	간다 나는 밥을 먹으려.	O
3.	먹으려 간다 나는 밥을.	O

번호	문장	정상여부
4.	밥을 먹으려 간다 나는.	O
5.	나는 먹으려 간다 밥을.	O
6.	나는 간다 밥을 먹으리.	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 먹으리.	X
12.	밥을 나는 먹으려 간다.	O
13.	나는 밥을 간다 먹으리.	X
14.	간다 나는 먹으려 밥을.	O
15.	먹으려 간다 밥을 나는.	O
16.	밥을 먹으려 나는 간다.	O

이러한 특징은 Parsing, POS Tagging 부터 Language Modeling에 이르기까지 한국어 NLP를 훨씬 어렵게 만드는 이유 중에 하나입니다.

또한 접사가 붙어 같은 단어가 다양하게 생겨나기 때문에, 하나의 어근에서 생겨난 비슷한 의미의 단어가 정말 많이 생성됩니다. 따라서 이들을 모두 다르게 처리할 수 없기 때문에, 추가적인 segmentation을 통해서 같은 어근에서 생겨난 단어를 처리하게 됩니다. 이와 관련한 내용은 Preprocessing 챕터에서 다루도록 하겠습니다.

읽을거리:

- <http://zomzom.tistory.com/1074>
- <https://m.blog.naver.com/reading0365/221057575669>

띄어쓰기의 어려움

애초에 동양권에서는 띄어쓰기라는 것이 존재하지 않았고 근대에 들어와서 도입된 것이기 때문에, 띄어쓰기에 맞춰 발전 해 온 언어는 아닙니다. 따라서 띄어쓰기에 대한 표준이 계속 바뀌어 왔기 때문에, 사람마다 띄어쓰기를 하는 것이 다를 뿐더러, 심지어는 띄어쓰기가 아예 없더라도 해석이 가능하기도 합니다. 결국, 위에서처럼 추가적인 segmentation을 통해서 띄어쓰기를 정제(normalization) 해 주는 process가 마찬가지로 필요하게 됩니다.



Figure 16: 내동생 고기 vs 내동 생고기



Figure 17: 농협용인육가공 vs 농협 용인 육가공

평서문과 의문문의 차이

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

물론 한국어에서도 의문문을 나타낼 수 있는 접사가 있습니다 – 니. 따라서, 점심 먹었니라고 하면 굳이 물음표가 붙지 않더라도 의문문인 것을 알 수 있습니다. 하지만 많은 경우에 그냥 의문문과 평서문이 같은 형태의 문장을 띠는 것이 사실입니다. 따라서, 마침표나 물음표가 붙지 않을 경우에는 알 수가 없는 경우가 많습니다. 특히나, 음성인식의 결과물로 나오는 text의 경우에는 더욱더 어렵습니다.

주어 생략

영어는 기본적으로 특성상 명사가 굉장히 중요시 됩니다. 따라서 정말 특별한 경우를 제외하고는 주어가 생략되는 경우가 없습니다. 하지만 한국어는 동사를 중요시하기 때문에, 주어가 자주 생략됩니다. 인간은 context 정보를 잘 활용하여 생략된 정보를 메꿀 수 있지만, 컴퓨터는 할 수가 없습니다. 따라서 위의 평서문과 의문문의 예에서도 볼 수 있듯이 한국어는 주어가 생략 되었는데, 컴퓨터는 누가 점심을 먹었고 누구에게 점심을 먹었냐고 물어보는지 알 수가 없습니다. 따라서 기계번역을 비롯하여 문장의 정확한 의미를 파악하는데 상당히 어렵게 됩니다.

읽을거리: – <http://www.hani.co.kr/arti/society/schooling/261322.html>
– <https://namu.wiki/w/%EC%A3%BC%EC%96%B4%EB%8A%94%20%EC%97%86%EB%8B%A>

한자 기반의 언어

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

이와 같이 원래 한국어도 한자 기반의 언어이기 때문에, 한자의 조합으로 이루어진 단어들이 많습니다. 이러한 단어들은 각 글자가 의미를 지니고 있고 그 의미들이

합쳐져 하나의 단어의 뜻을 이루게 됩니다. 이것은 영어에서도 마찬가지입니다. “water”와 같이 잉글로색슨족의 언어가 기원인 단어가 아니라면, latin 기반의 단어들은 각기 뜻을 가진 sub-word들이 합쳐져서 하나의 단어의 의미를 이루게 됩니다.

하지만 문제는 한글이 한자를 대체하면서 생겨났습니다. 한자는 표어 문자입니다. 문자 하나당 하나의 단어를 나타냅니다. 읽는 소리는 같을 지라도, 형태와 그 뜻은 다릅니다. 하지만 이것을 표음 문자인 한글이 나타내게 되면서, 정보의 손실이 생겨버렸습니다. (사실 표음 문자가 가장 발달한 글자의 형태입니다.) 인간은 이러한 정보의 손실로 생겨난 모호성(ambiguity)의 문제를 context를 통해서 효과적으로 해소할 수 있지만, 컴퓨터는 그렇지 못합니다. 따라서 다른 언어에 비해서 중의성이 더 가중되어 버렸습니다.

Type Text

원문 저는
여기
한
가지
문제점이
있다고
생각합니다.
형태소에
따른 는
segmentation
한
가지
문제점
이
있
다고
생각
합니다

TypeText
coun□저
based□는
sub-□여기
word□한
segmentation
□문
제
점
□이
□있
□다고
□생각
□합니다
□.
—

Preprocessing 챕터에서 다루겠지만, 이렇게 마지막까지 subword level로 segmentation할 경우에, 더욱 중의성 문제가 가중되어 버립니다.

문제점(□□□)이라는 단어가 문(□, 물을 문) 제(□, 제목 제) 점(□, 점 점)이라고 각각 segmentation 되었습니다. 하지만 결제(□□)의 제(□, 건널 제)도 있고, 제공(□□)의 제(□, 끌 제)도 있을 겁니다. 그런데 neural network에서 제라는 token은 결국 embedding vector로 변환이 될 겁니다. 따라서, embedding vector는 제(□, 제목 제), 제(□, 건널 제), 제(□, 끌 제) 세가지 모두에 대해서 embedding을 하게 될 것이고 저 뜻의 중앙 방향으로 vector가 애매하게 embedding 될 것 입니다. 사실, 굳이 neural network로 가지 않더라도, traditional NLP에서도 헷갈릴 것 또한 자명한 사실입니다. # Recent Trends in NLP

Conquering on Basic NLP

이전에 다루었던 대로, 인공지능의 다른 분야에 비해서 NLP는 가장 늦게 빛을 보기 시작하였다고 하였지만, 여러 task에 deep learning을 적용하려는 시도는 많이 이루어졌고, 진전은 있었습니다. 2010년에는 RNN을 활용하여 language modeling을 시도[Mikolov et al.2010][Sundermeyer et al.2012]하여 기존의 n-gram 기반의 language model의 한계를 극복하려 하였습니다. 그리하여 기존의 n-gram 방식과의 interpolation을 통해서 더 나은 성능의 language model을 만들어낼 수

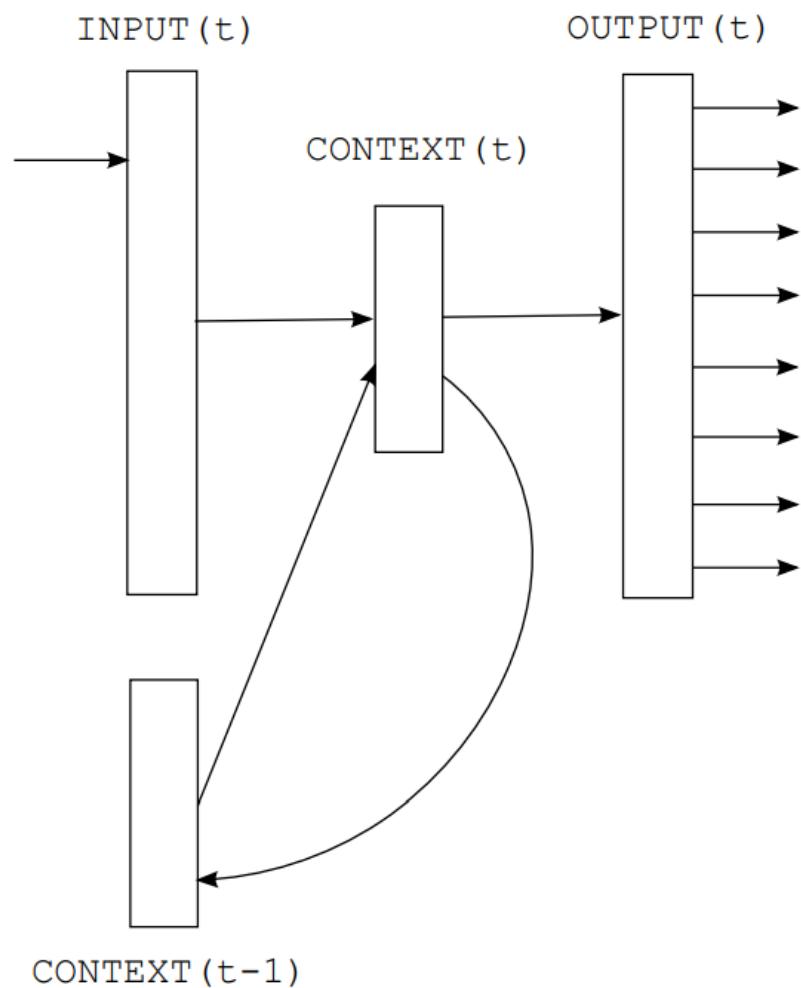


Figure 1: *Simple recurrent neural network.*

Figure 18:

있었지만, 기존에 language model이 사용되던 음성인식과 기계번역에 적용되기에는 구조적인 한계(Weighted Finite State Transducer, WFST의 사용)로 인해서 더 큰 성과를 거둘 수는 없었습니다. – 애초에 n-gram 기반 언어모델의 한계는 WFST에 기반하였기 때문이라고도 볼 수 있습니다. 닭이 먼저냐, 달걀이 먼저냐의 문제와 같음.

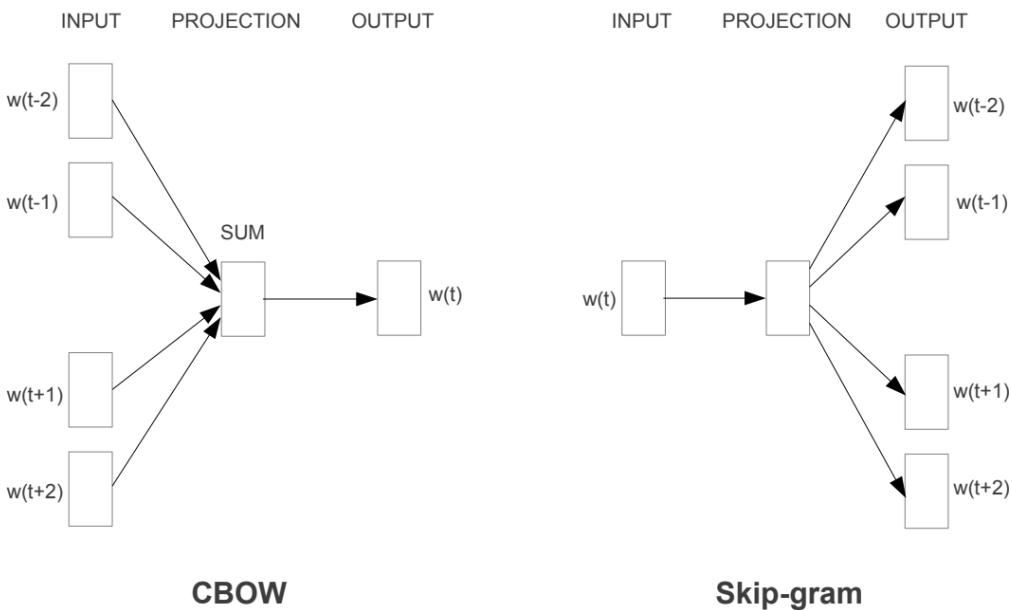


Figure 19:

그러던 와중에 Mikolov는 2013년 Word2Vec[Mikolov et al.2013]을 발표합니다. 단순한 구조의 neural network를 사용하여 효과적으로 단어들을 hyper plane(또는 vector space)에 성공적으로 projection(투영) 시킴으로써, 본격적인 NLP 문제에 대한 딥러닝 활용의 신호탄을 쏘아 올렸습니다. 아래와 같이 우리는 고차원의 공간에 단어가 어떻게 배치되는지 알 수 있음으로 해서, deep learning을 활용하여 NLP에 대한 문제를 해결하고자 할 때에 network 내부는 어떤식으로 동작하는지에 대한 insight를 얻을 수 있었습니다.

이때까지는 문장이란 단어들의 time series이기 때문에, 당연히 Recurrent Neural Network(RNN)을 통해 해결해야 한다는 고정관념이 팽배해 있었습니다 – Image=CNN, NLP=RNN. 하지만 2014년, Kim은 CNN만을 활용해 기존의 Text Classification보다 성능을 끌어올린 방법을 제시[Kim et al.2014]하며 한차례 파란을 일으킵니다. 이 방법은 word embedding vector와 결합하여 더 성능을 극대화 할

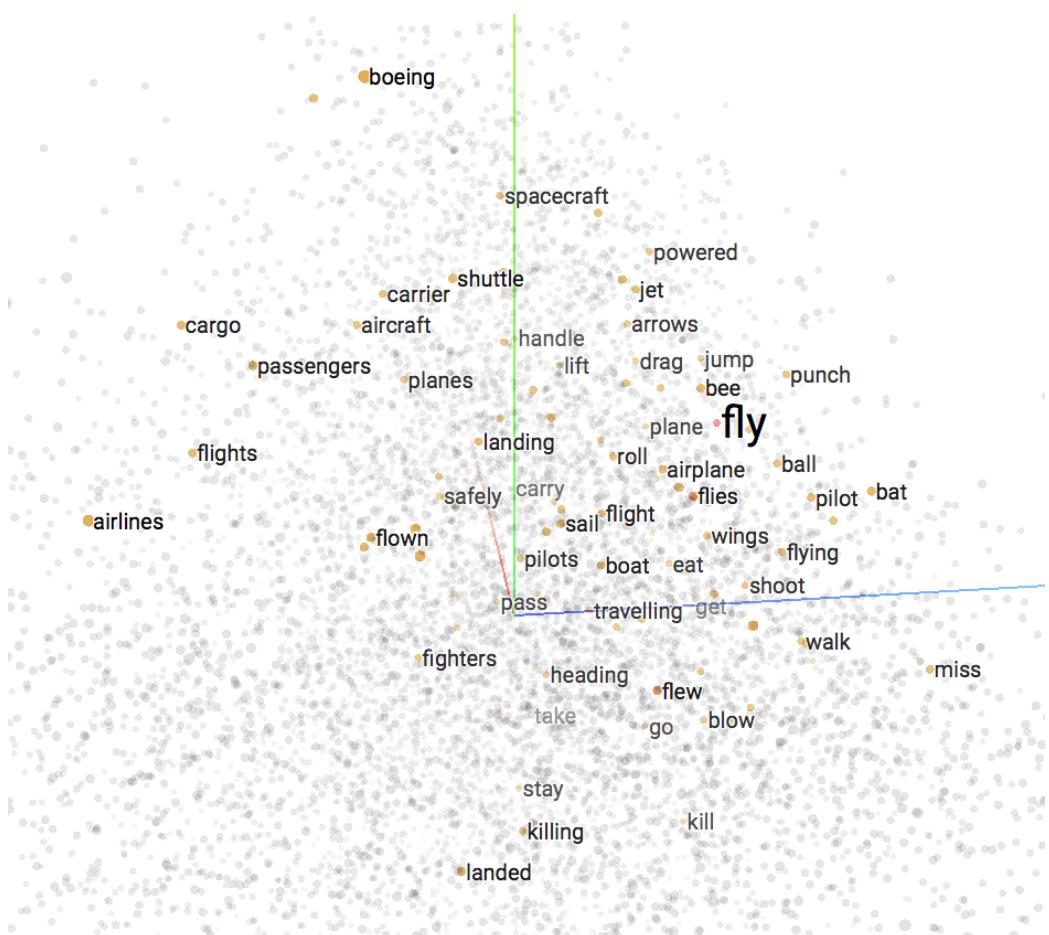


Figure 20:

수 있었습니다. 위의 paper를 통해서 학계는 NLP에 대한 시각을 한차례 더 넓힐 수 있게 됩니다.

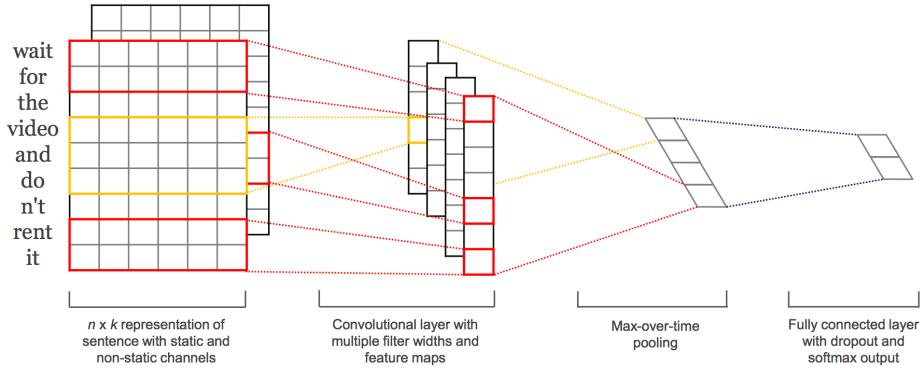


Figure 21:

이외에도 POS(Part-of-Speech) tagging, Sentence parsing, NER(Named Entity Recognition), SR(Semantic Role) labeling 등에서도 기존의 state of the art를 뛰어넘는 성과를 이루냅니다. 하지만 딥러닝의 등장으로 인해 대부분의 task들이 end-to-end를 통해 문제를 해결하고자 함에 따라, (또한, 딥러닝 이전에도 이미 매우 좋은 성과를 내고 있었거나, 딥러닝의 적용 후에도 큰 성능의 차이가 없음에) 큰 파란을 일으키지는 못합니다. – 당연히 그정도는 좋아지는거 아니야? 이런 느낌…?

Flourish of NLG

2014년 NLP에 큰 혁명이 다가옵니다. Sequence-to-Sequence의 발표[Sutskever et al.2014]에 이어, Attention 기법이 개발되어 성공적으로 기계번역에 적용[Bahdanau et al.2014]하여 큰 성과를 거둡니다. 이에 NLP분야는 일대 혁명을 맞이합니다. 기존의 한정적인 적용 사례에서 벗어나, 주어진 정보에 기반하여 자유롭게 문장을 생성할 수 있게 된 것입니다. 따라서, 기계번역 뿐만 아니라, summarization, 챗봇 등 더 넓고 깊은 주제의 NLP의 문제를 적극적으로 해결해보려 시도 할 수 있게 되었습니다.

또한, 이와 같이 NLP 분야에서 딥러닝을 활용하여 큰 성과를 거두자, 더욱더 많은 연구가 활기를 띠게 되어 관련한 연구가 쏟아져 나오게 되었고, 기계번역은 가장 먼저 end-to-end 방식을 활용하여 상용화에 성공하였을 뿐만 아니라, Natural Language Processing에 대한 이해도가 더욱 높아지게 되었습니다.

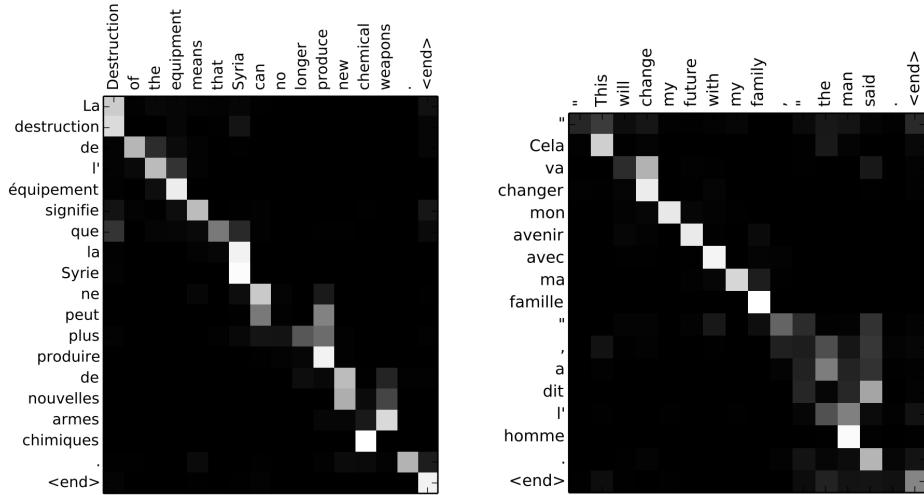


Figure 22:

Advanced Technique with Memory

Attention이 큰 성공을 거두자, continuous한 방식으로 memory에 access하는 기법에 대한 관심이 커졌습니다. 곧이어 Neural Turing Machine(NTM)[Graves et al.2014]이 대담한 이름대로 큰 파란을 일으키며 주목을 받았습니다. Continuous한 방식으로 memory에서 정보를 read/write하는 방법을 제시하였고, 이어서 Differential Neural Computer (DNC)[Graves et al.2016]가 제시되며 memory 활용방법에 대한 관심이 높아졌습니다.

이러한 memory를 활용하는 기법은 Memory Augmented Neural Network(MANN)이라 불리우며, 이 기법이 발전한다면 최종적으로는 우리가 원하는 정보를 neural network 상에 저장하고 필요할 때 잘 조합하여 꺼내쓰는, Question Answering (QA) task와 같은 문제에 효율적으로 대응 할 수 있게 될 것입니다.

참고사이트:

- <https://jamiekang.github.io/2017/05/08/neural-turing-machine>
- <https://sites.google.com/view/mann-emnlp2017/>

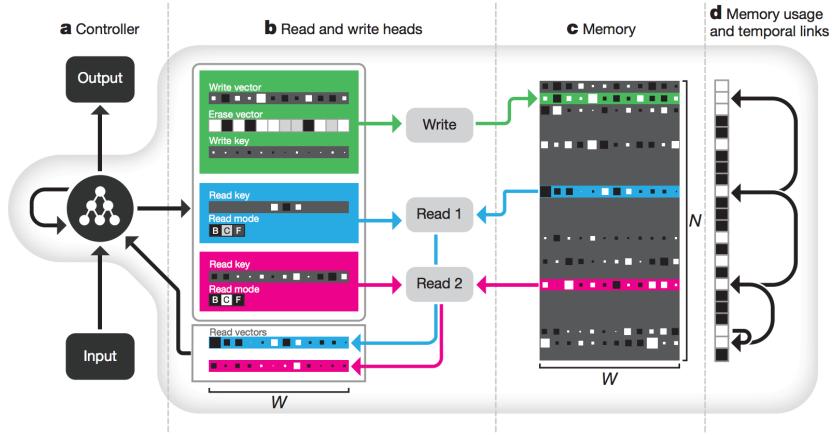


Figure 23:

Convergence of NLP and Reinforcement Learning

일찌감치 Variational Auto Encoder(VAE)[Kingma et al.2013]와 Generative Adversarial Networks(GAN)[Goodfellow et al.2014]을 통해 Computer Vision 분야는 기존의 discriminative learning 방식을 벗어나 generative learning에 관심이 옮겨간 것과 달리, NLP분야는 그럴 필요가 없었습니다. 이미 language modeling 자체가 문장에 대한 generative learning이기 때문입니다.

하지만, 기계번역의 연구결과서 큰 성과를 띠면서 학계는 다른 어려움에 부딪히게 됩니다. Deep learning에서 사용하는 cross entropy와 실제 기계번역을 위한 objective function과 괴리(discrepancy)가 있었기 때문입니다. 따라서, 마치 Computer Vision에서 기존의 MSE loss의 한계를 벗어나기 위해 GAN을 도입한 것처럼, 기존의 loss function과 다른 무엇인가가 필요하였습니다.

이때 성공적으로 강화학습의 policy gradients 방식을 NLP에 적용함으로써[Bahdanau et al.2016][Yu et al.2016], 마치 vision분야의 adversarial learning을 NLP에서도 흉내낼 수 있게 되었습니다. 이렇게, 강화학습(RL)을 사용하여 실제 task에서의 objective function으로부터 reward를 받을 수 있게 됨에 따라, 더욱 성능을 극대화 할 수 있게 되었습니다.

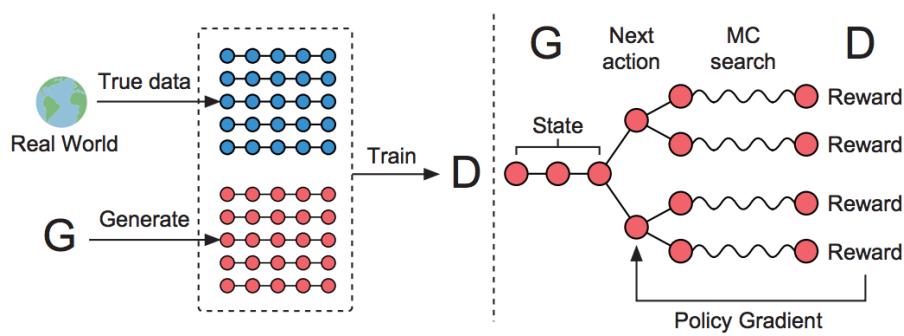


Figure 24: