

# Natural Language Processing with DeepLearning

## week 17

### The course

1. Large LM and GPT-3
2. Compositional Representations and Systematic Generalization
3. Improving how we evaluate models in NLP
4. Grounding language to other modalities

### 1. GPT-3

#### ① GPT-1

Semi-supervised : Unsupervised pre-training + supervised fine-tuning

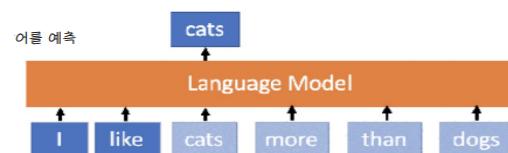
##### 1. Pre-train => Unsupervised

: Transformer decoder 를 이용해 대량의 레이블이 없는 코퍼스로 LM 을 사전학습

##### 2. Finetune => supervised

: pretrain 된 모델을 task 에 맞게 input 과 label 로 구성된 코퍼스에 대해 지도학습을 진행

Improving Language understanding by generative pre-training

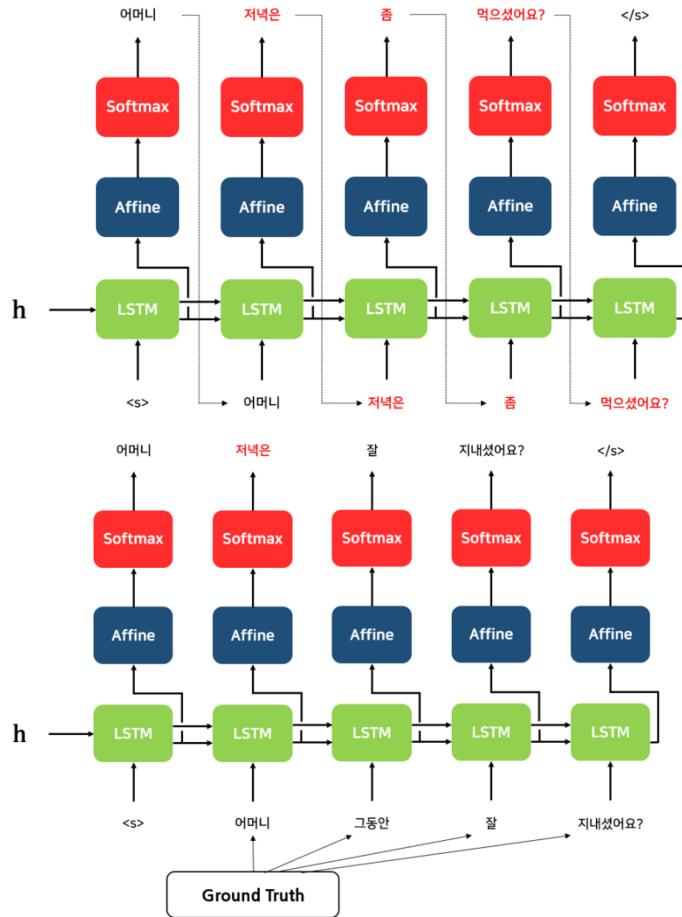


$$L = \sum_i \log P(w_i | w_0, \dots, w_{i-1}; \theta)$$

$$h_0 = UW_e + W_p$$

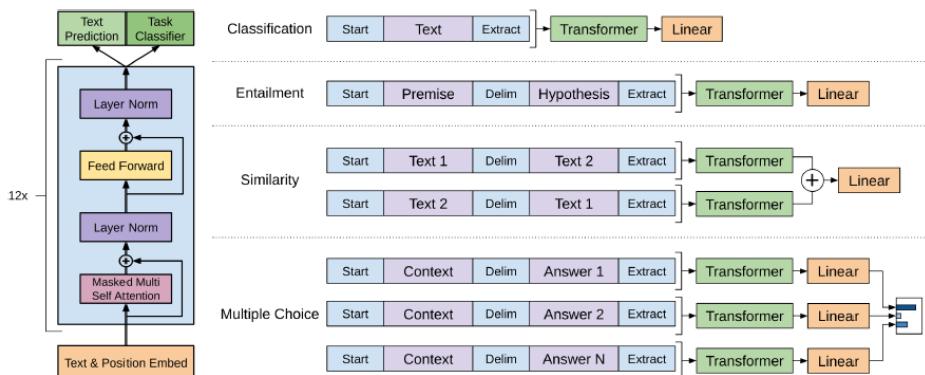
ls)  $h_l = \text{Decoder block}_l(h_{l-1})$   
 $P(w_i) = \text{softmax}(h_n W_e^T)$

- 트랜스포머 디코더만 사용
- 단방향 모델  $P(w_i) = P(w_i | w_0, \dots, w_{i-1})$  ↗ 이전까지 주어진 단어들로 현재 단어를 예측
- 일반적인 LM 을 통해 pre-train 을 진행
- LM 은 레이블이 필요 없다 ↗ 대량의 데이터 확보 가능
- Train data size : BooksCorpus (800M words)
- Teacher Forcing 을 이용 : target word(Ground Truth)를 디코더의 다음 입력으로 넣어주는 기법



(왼쪽 - 기존 학습 방법), (오른쪽 - teacher forcing 방법)

## ✓ Task



Transformer decoder로만 이루어짐

- Classification : 긍/부정, 문법 오류 여부
- Entailment : 주어진 문장들의 관계 분류
- Similarity : 두 문장 간 의미적 유사도 파악
- Multiple Choice : 주어진 문제에 대한 보기 중 정답 고르기

pre-train은 LM으로 진행되었으므로 각 task와 input 모양이 다를 수 있다는 문제점 존재

→ 각 task의 input을 GPT-1에 넣을 수 있도록 input을 위와 같은 모양으로 변형함  
(파랑보라 부분)

## ② GPT-2

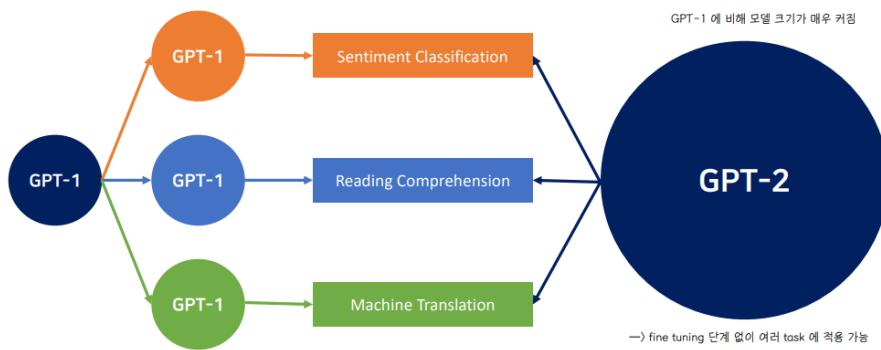
### ✓ LM are Unsupervised Multitask Learners

fine tuning 단계가 더 이상 필요없어진 LM👉 범용적인 LM을 만들자

- 기본 구조는 GPT-1과 동일
- zero shot learning : 모델이 바로 downstream task에 적용

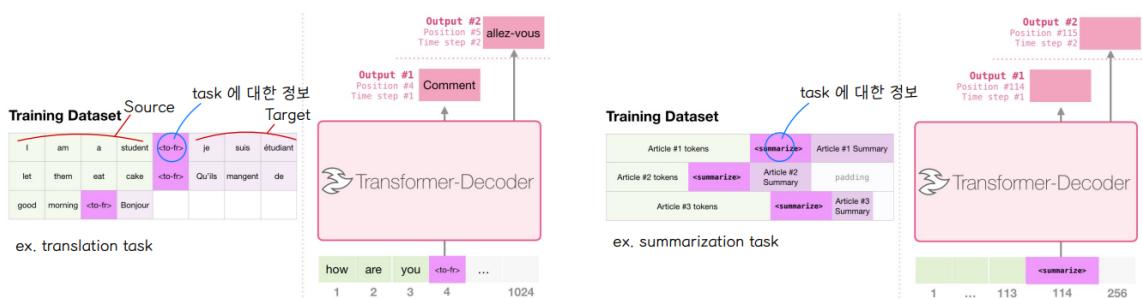
\*One-shot learning : downstream task를 한 건만 사용

\*Few-shot learning : downstream task를 몇 건 사용

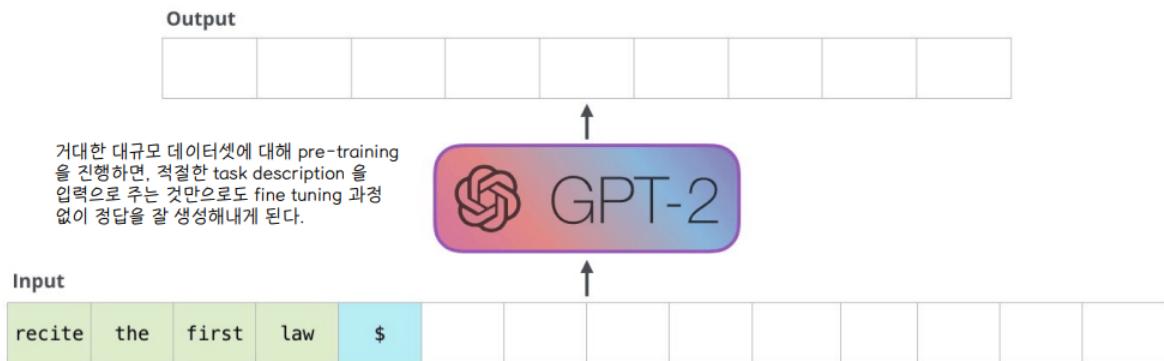


HOW ?

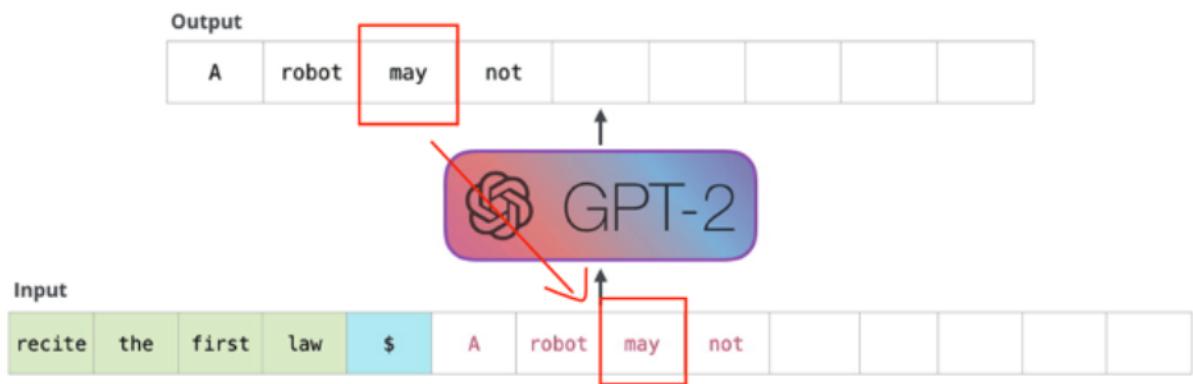
- task 정보를 함께 입력으로 넣어준다.



- 적절한 task description을 입력으로 함께 넣어주면 fine tuning 과정 없이 정답을 잘 생성해내게 된다.



### ✓ Auto regressive



- 이전 output 이 다음 input 으로 들어감

**✚ Byte pair encoding 활용 :** 글자와 단어의 중간 단위 subword 를 사용할 수 있음 , OOV 문제 해결

### ③ GPT-3

#### Open-AI 가 제작한 대형언어모델

대형언어모델은 또한 다양한 분야에서 실용적으로 활용될 수 있다. 대형언어모델을 이용하면 더 유창하게 대화할 수 있는 챗봇을 만들 수 있고, 문장이나 구절 몇 개만 주어지면 어떤 것에 관해서든 기사와 이야기를 만들어낼 수 있으며, 텍스트를 요약하거나 질문에 답을 하는 것도 가능하다. GPT-3를 이용하는 사람들은 이미 GPT-3를 이용해 창업 아이디어를 만드는 툴부터 던전을 배경으로 AI가 자유롭게 시나리오를 만드는 텍스트 어드벤처 게임에 이르기까지 수십 개의 다양한 앱을 만들고 있다.

GPT-3은 자연어 처리(NLP) 인공지능(AI)으로, 딥러닝을 사용해 인간과 같은 글을 생산해 내는 최신 언어 모델입니다.

언어 모델이란, 가장 자연스러운 단어의 연속 조합을 찾아내는 모델입니다. 어떤 한 단어 다음에 등장할 또 다른 단어를 예측하는 알고리즘이라고 보면 되겠습니다. 이 예측력이 뛰어날수록 인간과 기계의 언어는 가까워지게 됩니다. 두 번째로 자연어 처리란, 컴퓨터 프로그래밍처럼 인공적으로 만들어진 언어가 아닌 사람과 사람 사이 실제 사용하는 언어를 분석·처리하는 기술을 의미합니다. 이를 위해 머신러닝, 딥러닝 등 인공지능 기술을 활용하게 됩니다.

GPT-3의 이름에 숫자 3이 붙은 이유는 이 AI는 GPT(Generative Pre-trained Transformer)의 3세대 모델이기 때문인데요. GPT-3을 만든 오픈 AI에 따르면, GPT-3은 세계에서 가장 규모가 큰 언어 처리 모델로 GPT-2보다 최대 100배나 큰 약 1750억 개의 매개변수가 사용됐습니다. 입력된 단어만 해도 4990억 건에 이르는데요, 350기가바이트의 GPU 메모리까지 동원되었습니다. 평범한 클라우드 플랫폼에서 이 정도 자원으로 훈련을 시키려면 대략 300년이 넘게 걸린다고 하니 실로 어마어마한 학습량이 아닐 수 없습니다.

GPT-3은 자연어 기반 AI에게 대화, 통역, 번역 등의 기능을 기본적으로 수행합니다. 자유 대화의 경우 비록 장시간 맥락이 이어지는 수준의 대화는 아니지만 문학, 역사, 게임부터 시사에 이르는 다채로운 주제들에 대해 기존 AI보다 훨씬 자연스럽게 대화가 가능합니다. 심지어 GPT-3의 데이터 세트는 2019년 10월이 마지막이나 상황을 설명해 주면 그 이후의 사건에 대해서도 대화가 가능합니다. 아래는 GPT-3에게 코로나19 상황을 설명해 주고 나눈 대화 내용입니다.

또, GPT-3은 데이터 파싱(parsing)도 가능합니다. 데이터 파싱이란, 문자 데이터 내에서 목적에 맞는 데이터를 특정 패턴과 순서에 맞게 골라 가공하는 능력을 말합니다. 데이터 파싱을 수행하기 때문에 텍스트 요약이 가능하며 맥락에 맞되 보다 과장된 표현을 쓰라는 명령 수행이 가능합니다. 이 명령을 하면 GPT-3은 단순히 기존 단어를 재배열하는 수준을 넘어 각종 단어를 바꾼 다음 완전히 새로운 문장을 만들 어내며 예시를 직접 보여주면 성능이 높아지는 경향을 보였다고 합니다.

## ✓ LM are Few shot learners

- task에 대한 정보를 입력
- GPT-2와 달리 특별히 몇 가지 예제를 입력으로 넣어줌👉 Few Shot

GPT-2처럼 input에 해당 데이터가 어떤 task인지 특별한 token으로 명시해주는 것은 동일하나, GPT-2와는 달리 특별히 몇 가지 예제를 입력으로 넣어준다는 것에 차이가 있다. (Few Shot)

```
[example] an input that says "search" [toCode] Class App extends React Component... </div> } }  
[example] a button that says "I'm feeling lucky" [toCode] Class App extends React Component...  
[example] an input that says "enter a todo" [toCode]
```



GPT-3



## ✓ Large LM

- 엄청나게 많은 양의 데이터를 학습 시켰으며 크기가 매우 크다.

- 175 billion = 175,000,000,000 parameters
- Trained on 500 billion tokens

Model Name	$n_{\text{params}}$	$n_{\text{layers}}$	$d_{\text{model}}$	$n_{\text{heads}}$	$d_{\text{head}}$	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	$6.0 \times 10^{-4}$
GPT-3 Medium	350M	24	1024	16	64	0.5M	$3.0 \times 10^{-4}$
GPT-3 Large	760M	24	1536	16	96	0.5M	$2.5 \times 10^{-4}$
GPT-3 XL	1.3B	24	2048	24	128	1M	$2.0 \times 10^{-4}$
GPT-3.2.7B	2.7B	32	2560	32	80	1M	$1.6 \times 10^{-4}$
GPT-3.6.7B	6.7B	32	4096	32	128	2M	$1.2 \times 10^{-4}$
GPT-3 13B	13.0B	40	5140	40	128	2M	$1.0 \times 10^{-4}$
<b>GPT-3 175B or "GPT-3"</b>	<b>175.0B</b>	<b>96</b>	<b>12288</b>	<b>96</b>	<b>128</b>	<b>3.2M</b>	<b><math>0.6 \times 10^{-4}</math></b>

가장 크기가 큰 모델

엄청나게 많은 양의 데이터를 학습시킴

## ✓ Sparse attention pattern

- GPT-2 와 구조적으로 큰 차이는 없으나 attention 을 주는 부분에서 토큰 간 attention 을 전부 할당하면 계산량이 많아지는 것을 방지하기 위해 sparse 하게 attention 을 주는 부분에서 차이가 존재한다.
- Same architecture as GPT-2 with the exception of *locally banded sparse attention patterns*

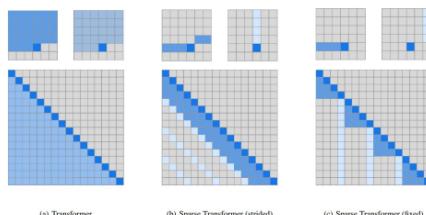
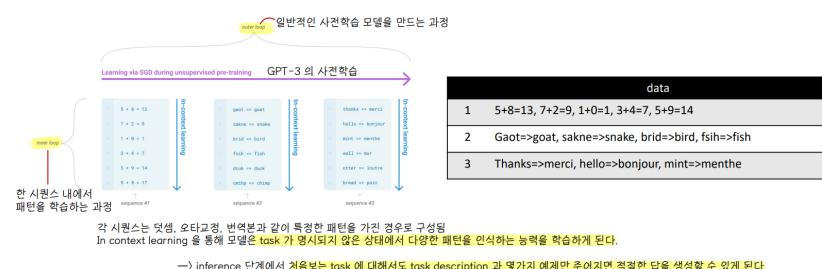


Figure 3. Two 2d factorized attention schemes we evaluated in comparison to the full attention of a standard Transformer (a). The top row indicates, for an example 6x6 image, which positions two attention heads receive as input when computing a given output. The bottom row shows the connectivity matrix (not to scale) between all such outputs (rows) and inputs (columns). Sparsity in the connectivity matrix can lead to significantly faster computation. In (b) and (c), full connectivity between elements is preserved when the two heads are computed sequentially. We tested whether such factorizations would match in performance the rich connectivity patterns of Figure 2.

Generating Long Sequences with Sparse Transformers (R Child et al, 2019)

## ✓ Meta-learning

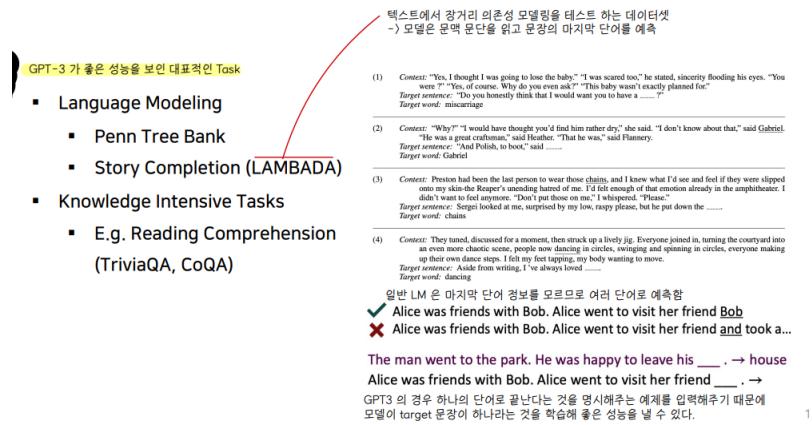


- 사람이 통제하던 기계학습 과정을 자동화하여 기계가 스스로 학습 규칙, 즉 메타 지식을 익힐 수 있게 하는 방법론
- 기계가 스스로 패턴 인식 능력을 개발 👉 inference 단계에서 원하는 task 에 빠르게 적응할 수 있음

- task 가 명시되지 않은 상태에서 다양한 패턴을 인식하는 능력을 학습하여, 처음보는 task 에 대해서도 task description 과 몇 가지 예제만 주면 적절한 답을 생성할 수 있게 된다.

## ✓ Task

### 좋은 성능을 보인 task



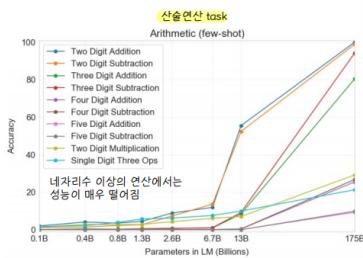
### Story completion

### 별로인 성능을 보인 task

| 모든 task 에서 엄청난 성능을 보인 것은 아니다.

- Structured problems that require multiple steps of reasoning
  - RTE, Arithmetic, Word problems, Analogy making

여러 논리적 추론을 요구하는 task 에서는 한계를 보임



논리적 추론을 요구하는 task 에서는 한계를 보임

## ✓ 한계점

- 翯 인터넷 자료들로 학습한 모델이므로 온라인의 수많은 가짜 정보, 편견을 그대로 학습
- 翯 인간과 같은 일반화 능력은 아직 도달하지 못함
- 翯 단순히 글로만 언어를 학습함 : merely learning from text without being exposed to other modalities
- 翯 언어모델을 구동하는데 필요한 엄청난 전력 소비

## 2. Compositional representation and systematic generalization

### ① 용어정리

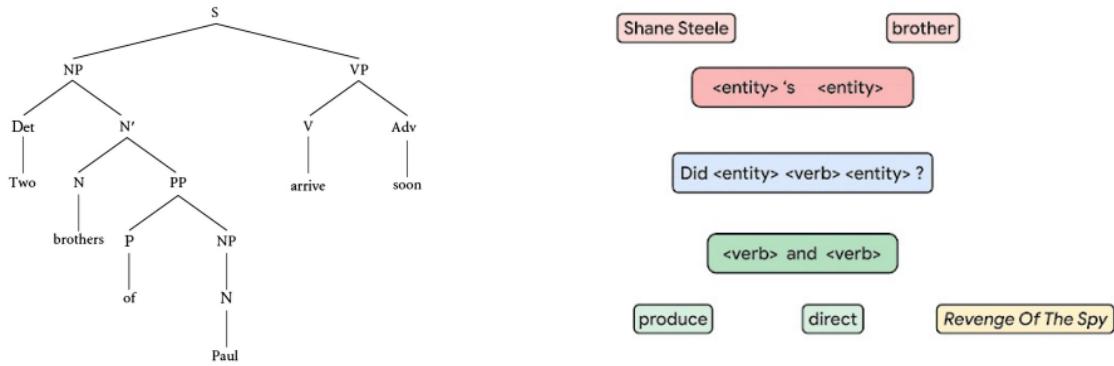
## ✓ systematicity

- 사람이 이해하는 문장들 간에는 확실하고 예측 가능한 패턴이 있다.

- EX. 철수는 영희를 좋아한다 → 영희는 철수를 좋아한다👉 인간의 언어 행동은 체계성을 갖추었으므로 앞의 문장을 이해했다면 뒤의 문장도 만들어낼 수 있다.

✓ compositionality

- 한 표현의 의미는 그 표현을 구성하는 구성 요소들의 의미와 구조로 구성된다.



## ② Compositional

✓ 인간의 언어

산 넘어 마을 = 넘다(산, 도착지)

구성성을 갖춘 경우



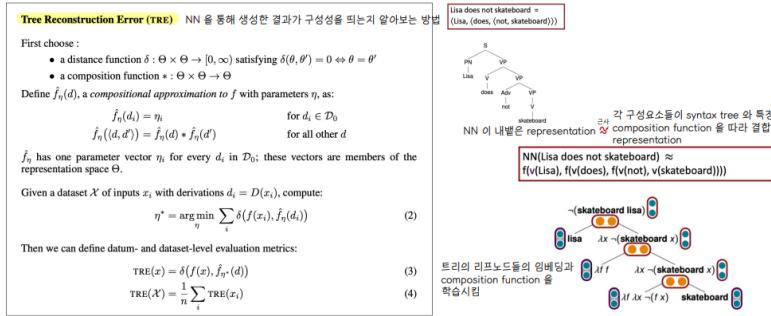
산 넘어 산 ≠ 넘다(산, 도착지)

그렇지 않은 경우



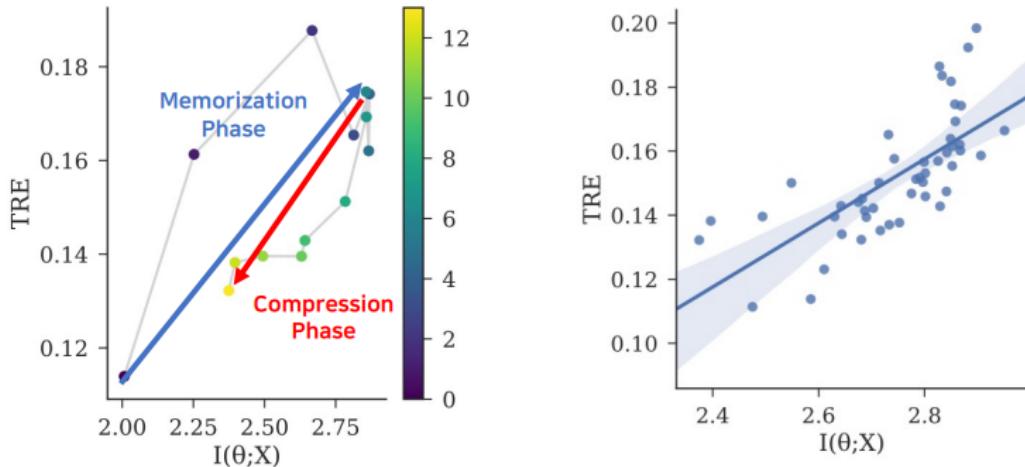
- 구성성을 갖춘 경우도 있고 그렇지 않은 경우도 있다.

✓ 신경망 표현



Measuring Compositionality in Representation Learning (Jacob Andreas, ICLR 2019)

- NN을 통해 생성한 결과가 구성성을 띠는지 확인하는 방법에 대한 연구
- NN이 결과로 내놓은 representation과, syntax tree와 composition function을 따라 결합된 representation를 근사시키어 구성성을 반영하도록 학습한다.



Measuring Compositionality in Representation Learning (Jacob Andreas, ICLR 2019)

- 현존하는 모델들이 구성 일반화 능력을 갖추었는지, 이런 능력을 측정하기 위해선 데이터셋을 어떻게 넓어야 하는지에 관한 연구

### ③ 일반화 능력

#### ✓ Compositional Generalization

- 이미 알고 있는 요소들로 새로운 조합을 만들거나 만들어진 조합을 이해할 수 있는 능력

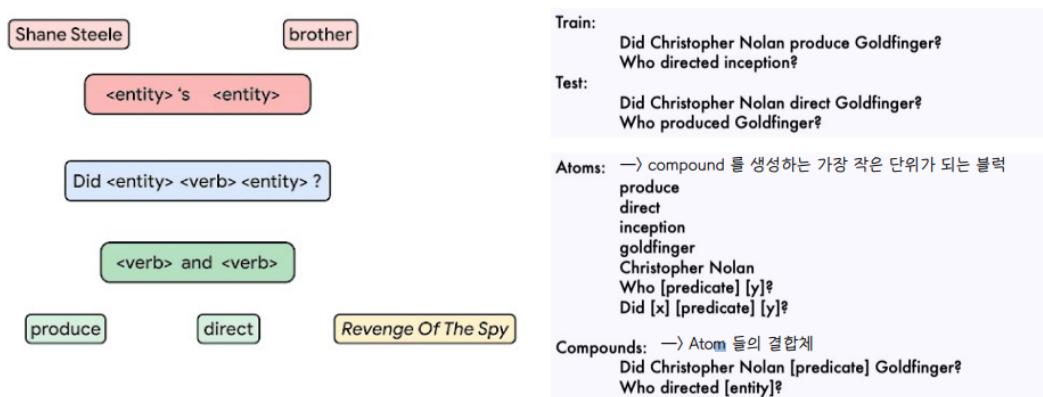
**Compositional Generalization** 구성 일반화 : 이미 알고 있는 요소들을 이용해 새로운 조합을 만들거나, 만들어진 새로운 조합을 이해할 수 있는 능력

- The capacity to understand and produce a potentially infinite number of novel combinations of known components. 일반적인 사람의 경우 새로운 단어를 하나 배우면 그 단어의 의미를 이용해 그 단어가 속한 새로운 단어의 조합을 쉽게 이해할 수 있음
  - 훼렉하다 → 밥을 훼렉하다, 훼렉하고 산책하다
- E.g. 모델이 알고 있는 단어 = [나, 사과, 먹다, 아침] 4가지 단어로 조합된 모든 문장을 이해할 수 있는 모델은 구성 일반화 능력이 좋은 모델임
  - 나 아침에 사과 먹었어, 아침에 나 사과 먹었어, 사과 먹었어 나 아침에, ...

## ✓ 관련 연구

모델의 구성 일반화 능력을 측정할 수 있는 방법을 제안한 논문에 대한 설명

[ai.googleblog.com](http://ai.googleblog.com)



Measuring Compositional Generalization: A Comprehensive Method on Realistic Data (Keysers et al, ICLR 2020)

## □ 모델의 구성 일반화 능력을 측정할 수 있는 방법을 제안한 논문

## ✓ 구성 일반화를 잘 반영하기 위한 데이터셋 분리 방법

□ Atom distribution 은 유사하면서 compound distribution 은 다르도록 dataset 을 split 하는 것이 가장 이상적임

👉 즉 사용되는 단어는 유사하나, 단어의 조합 형태는 다를수록 구성 일반화를 잘 반영한 것

- 이상적으로 조개는 방법
- Ideal Compositionality Experiment**

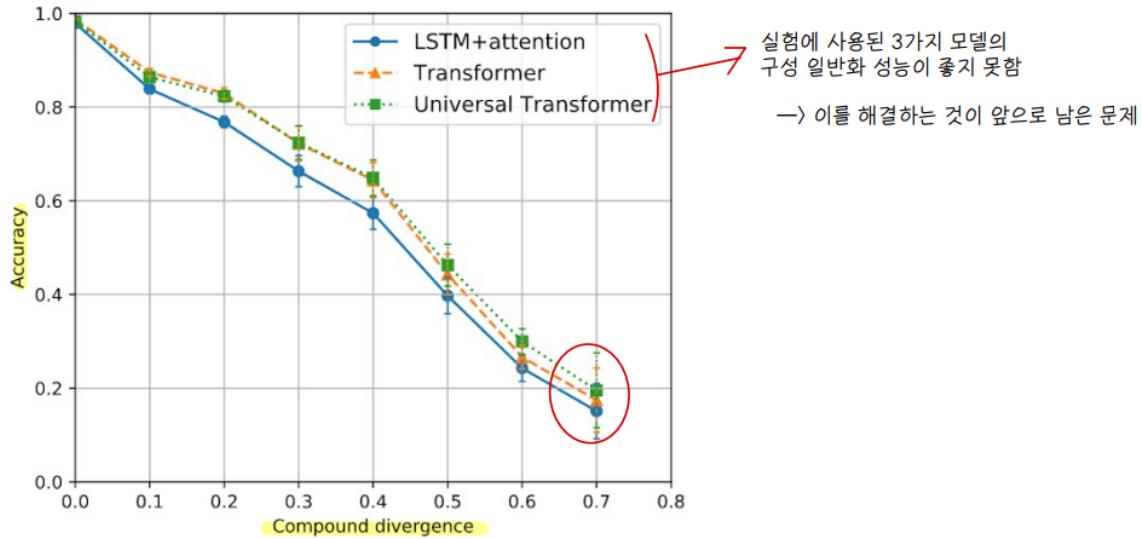
Atom distribution 은 유사하면서 compound distribution 은 다른 환경

  - Similar atom distribution: All atoms present in the test set are also present in the train set, and the distribution of atoms in the train set is as similar as possible to their distribution in the test set.
  - Different compound distribution: The distribution of compounds in the train set is as different as possible from the distribution in the test set.
- Split data into train / test such that compound divergence is maximized and atom divergence is minimized! → 이럴수록 구성일반화 능력을 잘 측정할 수 있는 데이터셋!

Train set	Test set	
Who directed Inception?	train set에서 자주본 단어들, 그러나 새로운 조합의 문장	→ test set에서 좋은 성능을 보이는 모델
Did Greta Gerwig produce Goldfinger?	Did Greta Gerwig direct Goldfinger?	= 좋은 구성 일반화를 갖춘 모델
...	Who produced Inception?	
... who, directed,inception 과 같은 atom 이 동일하게 나오지만, 해당 atom 이 조합된 형태는 다 다름 (=compound distribution 은 다름)		

## □ 구성 일반화 측정 실험

realistic language? atom divergence 는 최소로 설정하고 compound divergence 만 변경해보며 성능을 측정



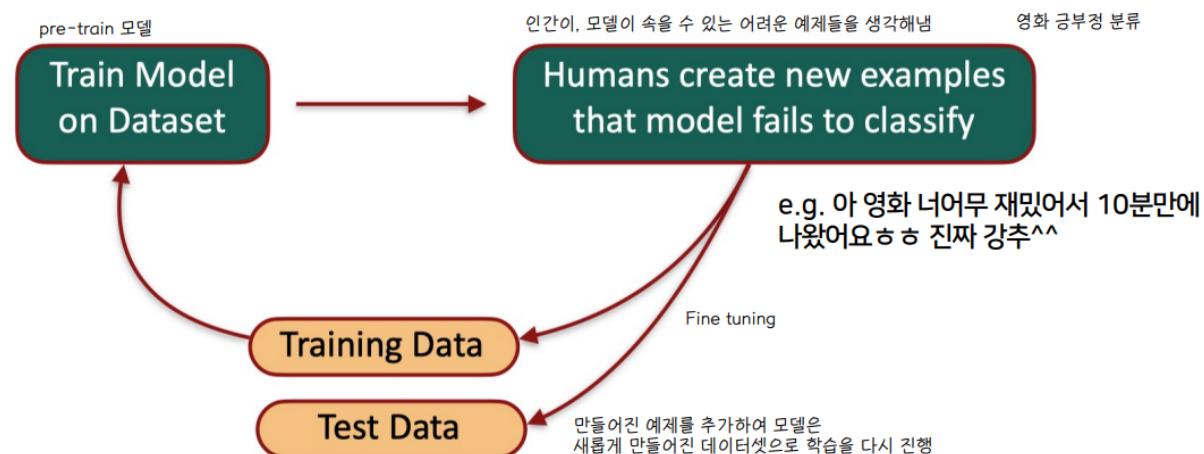
<i>Model</i>	<i>CFQ (Maximum Compound divergence)</i>	
T5-small (no pretraining)	21.4	compound divergence 를 최대화 시키고 모델의 scale 을 늘렸을 때 성능이 좋아지긴 하지만 scale 을 계속 늘린다고 계속 성능이 좋아질지 모르기 때문에 모델 크기를 늘리는게 방도는 아님
T5-small	28.0	
T5-base	31.2	
T5-large	34.8	
T5-3B	40.2	
T5-11B	40.9	
T5-11B-mod	42.1	

### ③ NLP model 의 성능 평가를 개선하는 방식

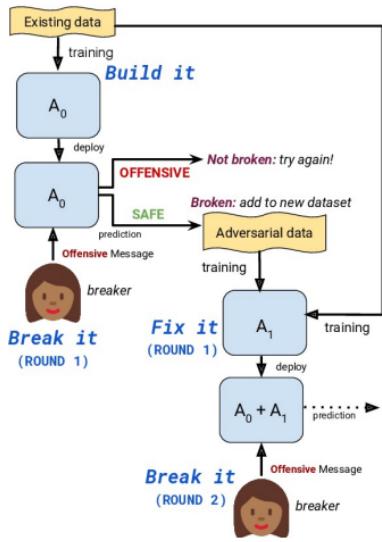
#### ① 모델의 성능

🤔 벤치마크 데이터셋에서의 성능은 날로 증가하지만, 실제 세계에서의 모델 성능도 그러한가

#### ✓ Dynamic benchmarks



👉 모델이 속을 수 있을 만한 어려운 예제들을 추가하여 학습을 진행하도록 함



준비한 데이터셋으로

1. **Build it** : 사용자의 공격적인 메세지를 감지할 수 있는 모델 개발
2. **Break it** : Crowdworker에게 모델은 "SAFE"하다고 생각하지만 Crowdworker는 "OFFENSIVE"하다고 생각하는 메세지를 만들어서 모델을 속여달라고 요청, "beat the system"해달라고 요청
3. **Fix it** : 2번 과정을 통해 모여진 예제들을 통해 모델을 재학습 → 적대적인 공격에 더 강건한 모델이 될 수 있도록!
4. **Repeat** : Break it - Fix it 을 계속계속 반복

기존 데이터셋에 추가해

인간과 모델이 상호작용하며 계속 반복

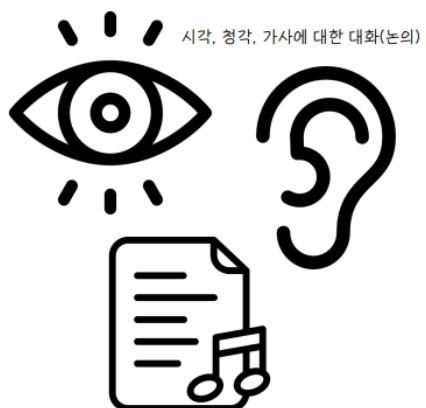
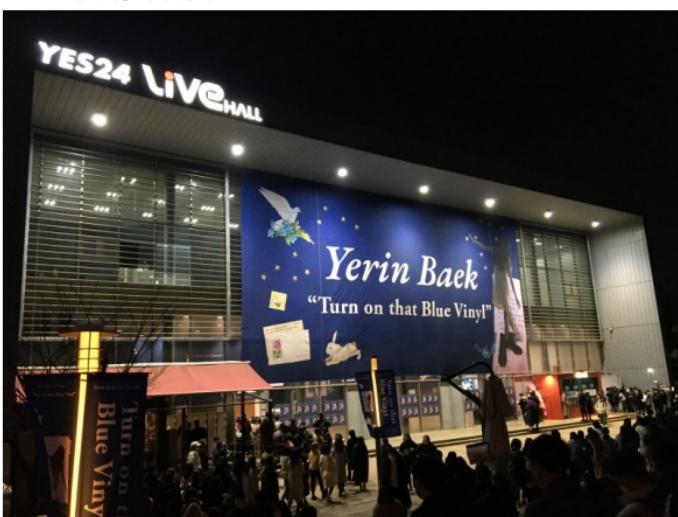
대화 시스템에서 다이나믹 벤치마크를 활용한 예시 논문

**Build-It Break-It Fix-It for Dialogue Safety (Dinan et al, EMNLP 2017)**

#### 4 Grounding language to other modalities

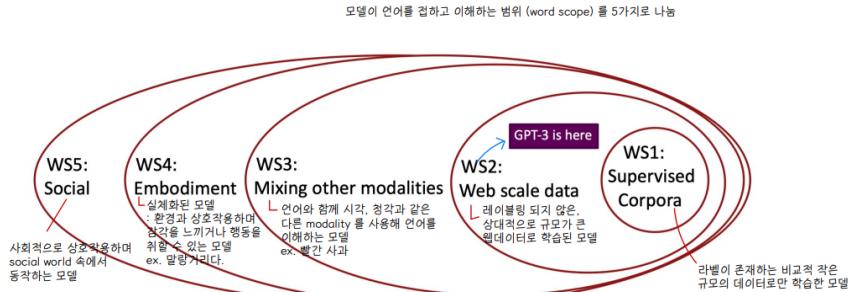
- ✓ 단순한 텍스트를 넘어 다양한 modality 를 사용해 언어를 이해하는 것

노래 가사를 이해하는 것



👉 텍스트로만 학습하는 것은 언어가 무엇에 관한 것인지 그것이 어떤 작용을 하는 것인지 알기 어렵다.

## ✓ Grounds language



Computer vision and speech recognition are mature enough for investigation of broader linguistic contexts (WS3). The robotics industry is rapidly developing commodity hardware and sophisticated software that both facilitate new research and expect to incorporate language technologies (WS4). Simulators and videogames provide potential environments for social language learners (WS5). Our call to action is to encourage the community to lean in to trends prioritizing grounding and agency, and explicitly aim to broaden the corresponding World Scopes available to our models.

컴퓨터 비전과 음성인식을 이용하면 모델이 더 넓은 언어적 맥락 속에서 언어를 이해할 수 있고 로봇산업에서 사용중인 HW, SW 를 사용하면 더 넓은 맥락 속에서 언어를 이해할 수 있게 된다. 시뮬레이터와 비디오 게임을 이용하면 더 넓게 이해할 수 있다.

단순히 모델이 텍스트만을 이용해 언어를 이해하는 것을 넘어서, 더 넓은 맥락 속에서 언어를 이해할 수 있도록 해야한다.

Experience Grounds Language (Bisk et al, EMNLP 2020)