GPT

박지훈, 김성환, 은하영

목차

1. NLP 모델들의 년도테이블
2. BERT와 GPT의 형식적 차이
3. Prerequisite Requirement For GPT
    a. Language Model
    b. Generative Model & Discriminative Model
    c. Supervised Learning & Unsupervised Learning
    d. Pre-Training & Fine-Tuning
    e. Auxiliary
    f. BPE(Byte-Per-Encoding)
4. Model Description
    a. Model Architecture
    b. Pre-Training
    c. Supervised Fine-Tuning
    d. Auxiliary Objectives
    e. Experiments
5. 추가적인 실험 [Analysis]
    a. Pre-Training의 층이 많을수록 효과적일까?
    b. Zero-shot Behaviors
6. GPT-1의 한계와 의의
7. GPT-2
8. GPT-3

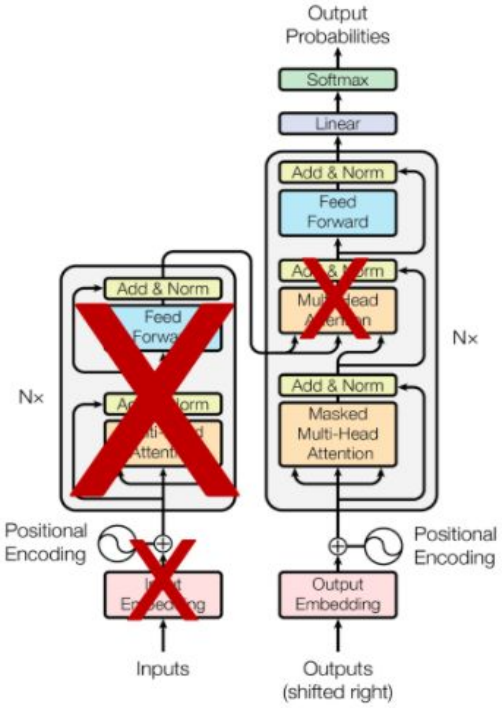
GPT-1 논문: https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf

1. NLP 모델들의 년도테이블

RNN(1986)	LSTM(1997)	Seq2Seq(2014)	Attention(2015)	Transformer(2017)
병렬화가 불가능, 길이가 길어질 수록 기울기 소실	시그모이드 곱으로 인한 메모리 삭제가능성, 연산속도 느림	입력 시퀀스를 하나의 벡터로 압축하는 과정에서 정보가 일부 손실	해당 시점에서 예측해야할 단어와 연관이 있는 입력 단어 부분을 좀 더 집중한다는 아이디어	seq2seq의 구조를 따르면서 attention만으로 구현

2. BERT와 GPT의 형식적 차이

	BERT	GPT
Model	Masked Language Model	Language Model
Transformer의 형식	Encoder	Decoder
Pre-train과정	문장 중간에 빈칸을 만들고, 해당빈칸이 어떤 단어가 적절할지 맞추는 과정	이전 단어들이 주어졌을때, 다음단어가 무엇인지 맞추는 과정
학습 방향	Bidirectional	Unidirectional



3. Prerequisite Requirement For GPT

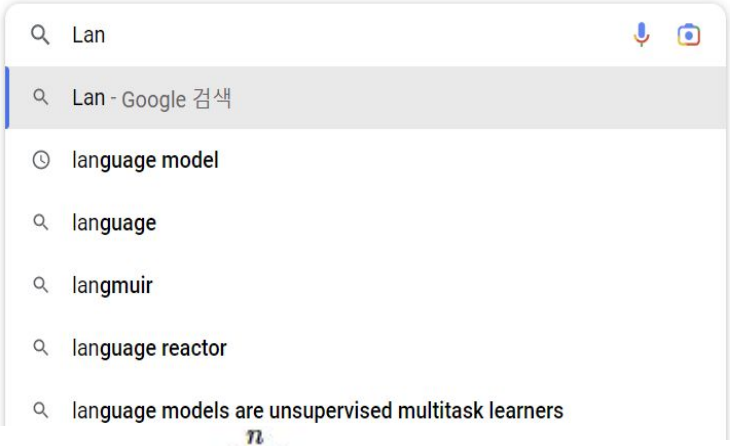
- ❏ Language Model
- ❏ Generative model & Discriminative model
- ❏ Supervised Learning & Unsupervised Learning
- ❏ Pre-Training & Fine-Tuning
- ❏ Auxiliary
- ❏ BPE(Byte-Per-Encoding)

Language Model

앞에 어떤 단어들이 나왔는지 고려하여 **후보가 될 수 있는 여러 단어들에 대해서 등장 확률을 추정**하고
가장 높은 확률을 가진 단어를 선택

LM이란?

언어라는 현상을 모델링하고자 단어 **시퀀스(문장)에 확률을 할당**하는 모델.



$$P(W) = P(w_1, w_2, w_3, w_4, w_5, \ldots, w_n)$$

w : 하나의 단어
W : 단어 시퀀스

다음 단어 등장 확률 ➡ $P(w_n|w_1, \ldots, w_{n-1})$

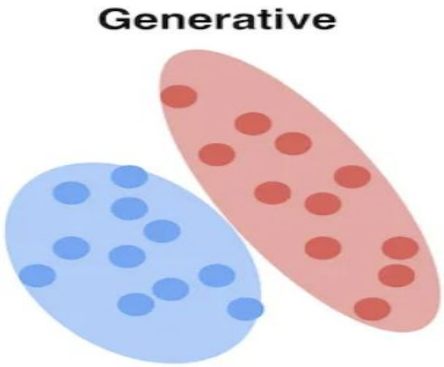
전체 단어 시퀀스W의 확률은 모든 단어가 예측되고 나서야 알 수 있다.

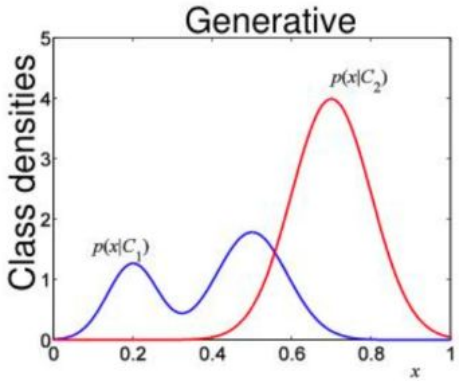
단어 시퀀스의 확률 ➡ $P(W) = P(w_1, w_2, w_3, w_4, w_5, \ldots w_n) = \prod_{i=1}^{n} P(w_i|w_1, \ldots, w_{i-1})$

Generative model & Discriminative model

Generative

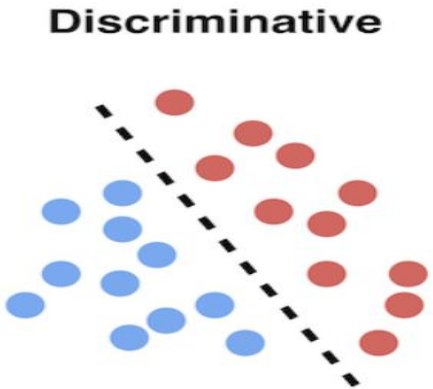
- 주로 unlabeled data
- 어떻게 분류될지에 대한 확률

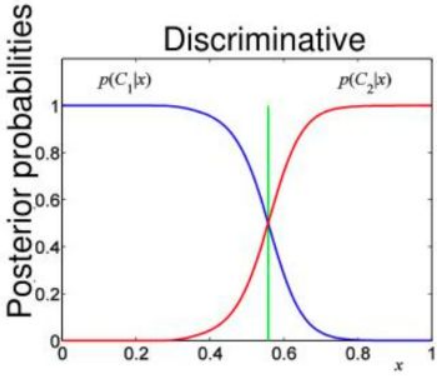




Discriminate

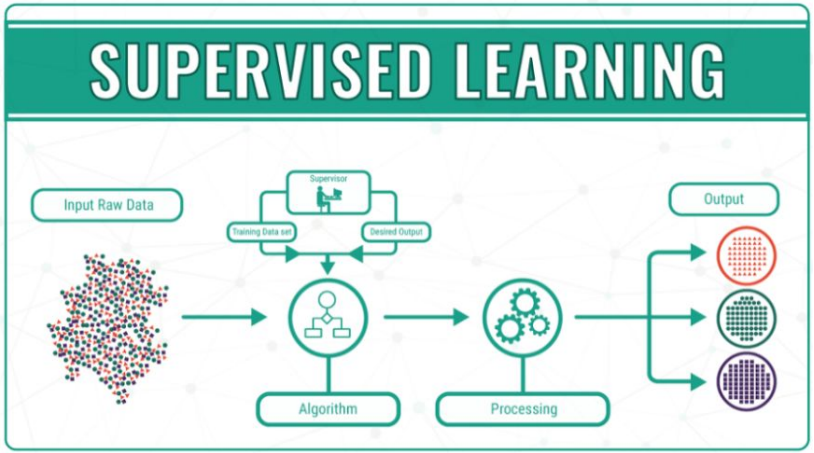
- Only labeled data
- 많은 데이터에서 더욱 효과적
- 승/패, 성공/실패





Supervised Learning & Unsupervised Learning

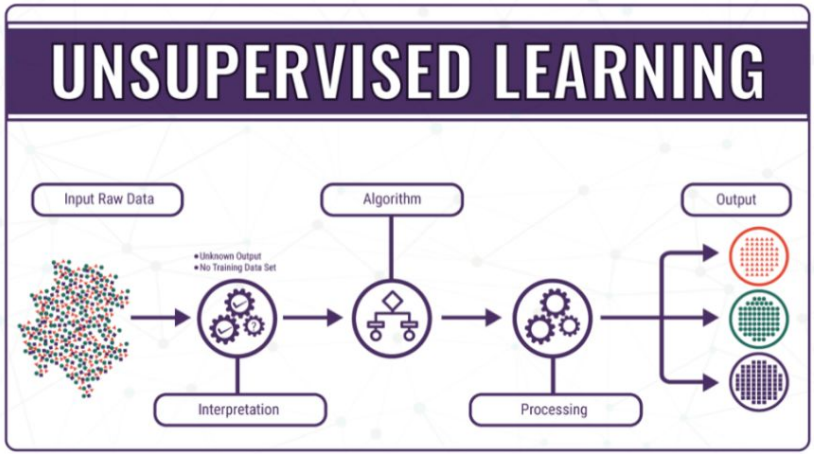
Supervised Learning



- **Training data로부터 하나의 함수를 유추**해내기 위한 기계학습의 한 방법이다.

- **Labeled data를 기반으로 학습**하여 알고리즘 모델을 만들고 **unlabeled data의 label을 예측**한다.

Unsupervised Learning



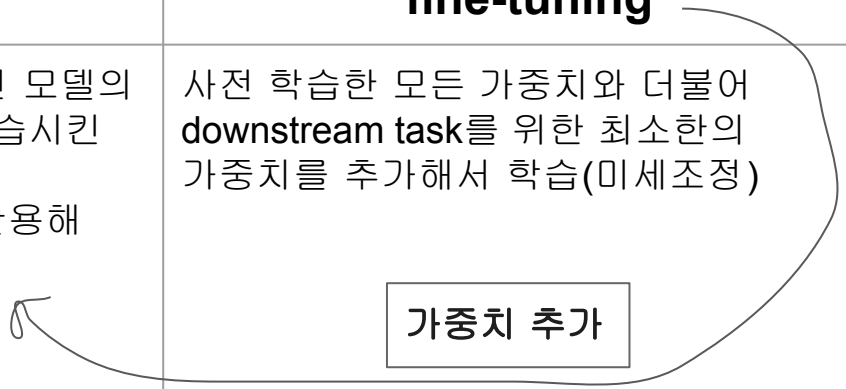
- **Unlabeled data에 대해 컴퓨터가 스스로 학습**하는 기계학습의 한 방법이다.

- **Unlabeled data로 특정한 패턴과 규칙의 알고리즘 모델을 만든다.**

즉, y없이 x만 이용해서 학습하는 것이다.

Pre-Training & Fine-Tuning

pre-training	fine-tuning
기존에 임의의 값으로 초기화하던 모델의 가중치들을 다른 문제(task)에 학습시킨 가중치들로 초기화 ex) 감정 분석 모델의 가중치를 활용해 텍스트 유사도 모델(다른 task)의 가중치로 활용	사전 학습한 모든 가중치와 더불어 downstream task를 위한 최소한의 가중치를 추가해서 학습(미세조정)



Auxiliary

Auxiliary training objectives Adding auxiliary unsupervised training objectives is an alternative form of semi-supervised learning. Early work by Collobert and Weston [10] used a wide variety of auxiliary NLP tasks such as POS tagging, chunking, named entity recognition, and language modeling to improve semantic role labeling. More recently, Rei [50] added an auxiliary language modeling objective to their target task objective and demonstrated performance gains on sequence labeling tasks. Our experiments also use an auxiliary objective, but as we show, unsupervised pre-training already learns several linguistic aspects relevant to target tasks.

Auxiliary training objectives

보조적으로 비지도 학습 목적함수를 추가하는 것 또한 준 지도학습의 형태 중 하나이다.

본 연구에서 auxiliary objective를 사용하지만, unsupervised pre-training에서 이미 target task와 관련 있는 언어적 특성이 충분히 학습된다.

Objective function

objective function(목적함수)은 **최적화 문제에서 값을 최소화하거나 최대화하는 것을 목적으로 하는 함수**를 의미한다.

반대로, objective function을 최대화 또는 최소화하는 인수를 구하는 문제가 최적화 문제이다.

Unsupervised pre-training

$$L_1(\mathcal{U}) = \sum_i \log P(u_i|u_{i-k}, \ldots, u_{i-1}; \Theta)$$

L1(U)은 Unsupervised pre-training을 통한 **language model**이다.

Supervised pre-training

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m).$$

GPT-1(Improving languague understanding by Generative Pre-Training)란?+벡터 흐름 하나하나 자세하게 설명

We additionally found that including language modeling as an auxiliary objective to the fine-tuning helped learning by (a) improving generalization of the supervised model, and (b) accelerating convergence. This is in line with prior work [50, 43], who also observed improved performance with such an auxiliary objective. Specifically, we optimize the following objective (with weight λ):

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C}) \quad (5)$$

fine-tuning에 auxiliary objective로써 Language Model을 추가하는 것을 auxiliary objective function이라고 한다. (준 지도학습의 한 형태)

L3는 성능향상을 위해 보조 목적함수를 더하는 식으로 학습을 수행한다.

fine-tuning에 보조 목적 함수로서 language model(=L1(C))을 추가한다.

* C=labeled data

Auxiliary objective function을 사용하였을 때 성능향상

1. supervised model의 **일반화 능력을 증가**시킨다. (모델을 변화에 강인하게)

2. 모델이 **빠르게 수렴**하는 데 도움을 준다.

그래서 최종적으로 다음 목적함수 L3(C)를 최적화한다.

* 연구자가 가중치 λ를 조정하여 최적화(하이퍼 파라미터)

즉, 본 논문에서 Auxiliary objective function은 λ*L1(C)이다.

L1은 라벨링 되지 않은 말뭉치의 토큰(단어)들의 순서정보를 뽑아내기 위해 k개의 이전 단어들에 대해 그다음 단어의 등장 가능성을 **최대화**하였고 이를 모든 n개의 단어에 대해 진행한다.

L2는 labeled dataset C를 pre-trained model에 집어넣고

마지막 transformer block의 마지막 토큰에 대한 activation인 hlm을 linear layer에 넣고

그에 대한 softmax 값을 라벨 y를 예측하는 데 사용한다.(=x의 input sequence가 주어짐)

그리고 그에 따른 확률을 높이는 목적함수(L2(C))를 **최대화**한다.

L3는

1. unsupervised pre-training 이후 모델이 준비되면

2. labeled dataset C를 가지고

3. L2(C)를 최대화하는 동시에

4. λ만큼의 비율로 L1(C) 또한 **최대화**하는 것이다.

결론: L1, L2, L3 모두 목적함수이다. / auxiliary 목적함수는 λ*L1(C)이다.

BPE(Byte-Per Encoding)

BPE란?

OOV의 문제를 해결하기 위한 새로운 인코딩 방법

기존의 인코딩 방법

1. 단어를 글자로 분리
2. 빈도수에 따라 점차적으로 결합 →**단어집합 생성**

예시 - 지코의 Tough Coockie

dic = {'low':5, 'lower':2, 'newest':6, 'widest':3} 이 있다고 가정.
'lowest'라는 단어는 의미가 비슷한 단어는 있으나, 존재하진 않으므로, OOV문제 발생!!

BPE - iteration

1. **서브워드 분리(subword segmentation)**

dic = [l o w l o w e r n e w e s t w i d e s t]
vocabulary = [l o w e r n w s t i d]

2. 가장 높은 빈도의 (e,s)쌍을 **es**로 통합

dic = [l o w l o w e r n e w **es** t w i d **es** t]
vocabulary = [l o w e r n w s t i d **es**]

3. 빈도수가 가장 높은 (es, t)쌍을 **est**로 통합

dic =[l o w l o w e r n e w **est** w i d **est**]
vocabulary = [l o w e r n w s t i d es **est**]

"lowest"
= "low" + "est"

4. Model Description

- ❏ Model Architecture
- ❏ Pre-Training
- ❏ Supervised Fine-Tuning
- ❏ Auxiliary Objectives
- ❏ Experiments

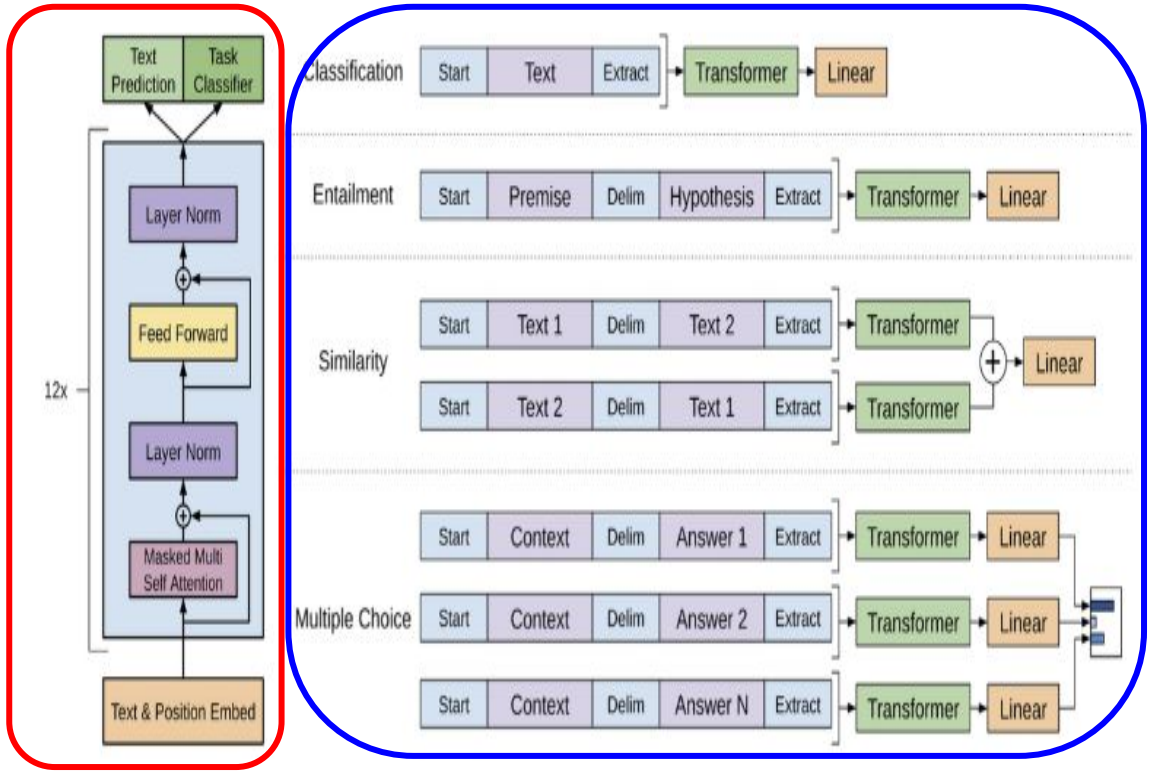
Model Architecture

GPT1은 2step으로 이루어져 있습니다.

1. **Unsupervised pre-training (unlabeled data)**

2. **Supervised fine-tuning (labeled data-목적에 맞게 미세조정)**

3. (Auxiliary objectives)



Pre-Training

트랜스포머의 Decoder구조만을 따르면서,
Masked-multi-head Attention만을 이용합니다.

이외의 과정은 Transformer의 decoder block과 동일하게 수행합니다.

1 $h_0 = UW_e + W_p$

input token Matrix U에 **embedding Matrix W_e**를 곱한 후, **Positional Embedding W_p**를 더해서, Masked Attention의 **input h_0**을 만듭니다.

2 $h_l = transformer\_block(h_{l-1}) \forall i \in [1, n]$

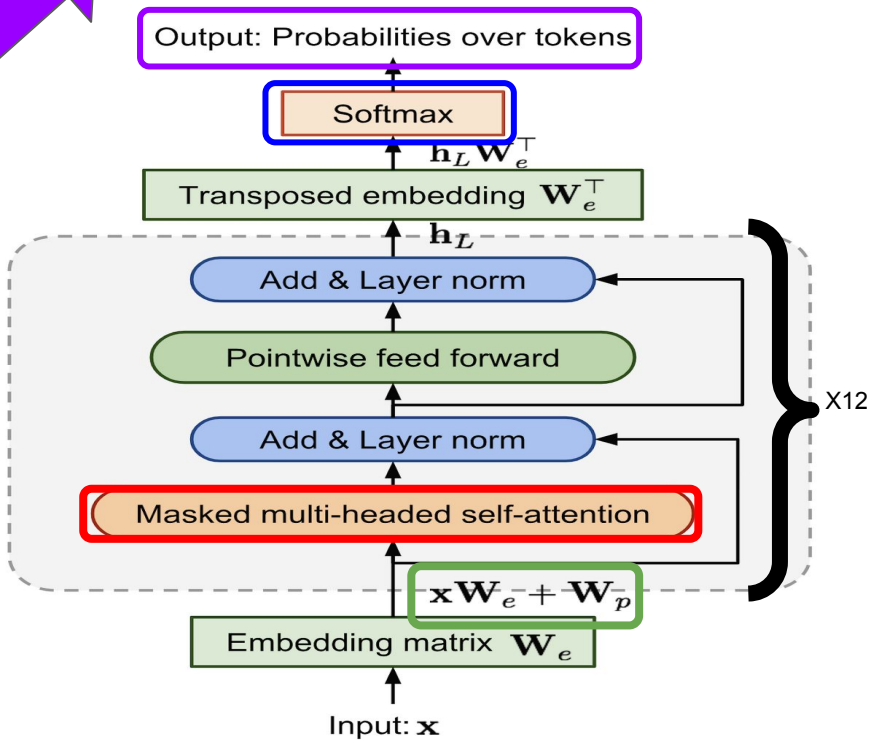
l번째 decoder block의 hidden state **h_l은 h_l-1를 입력으로 받아 계산**됩니다.

3 $P(u) = softmax(h_n W_e^T)$

마지막 h_n에 다시, **W_e.T**를 곱하여 **softmax**를 취하면, **output probability**를 구할 수 있습니다.

이 과정으로 모델링한 조건부확률 P(u)를 통해 unlabeled token u를 **L1(u)를 최대화하는 방식으로 학습**

$$L_1(u) = \sum_i logP(u_i|u_{i-k}, \ldots, u_{i-1}; \theta)$$



Supervised Fine-Tuning

pre-training이 끝난 후, **target task**에 맞게 **parameter를 조정하는 단계**입니다.

input token sequence와 **labeled dataset1(C)**를 통해 학습이 진행됩니다.

1. 우선 C에 대한 **input token 정보 (h_ml)**를 얻기 위해서, pre-trained model에 input token들을 통과시킵니다.
2. 그리고 파라미터Wy를 갖는 하나의 linear layer에 h_ml을 통과시켜 **softmax probability**를 계산합니다.
3. 이결과로 **token probability distribution**을 얻을 수 있고, **label y에 대해서 지도학습을 진행**할 수 있습니다.
4. 지도학습의 목적함수 **L2(C) likelihood 또한 일련의 구조로 모델링된 조건부확률 P를 통해 계산**된다.(**L2최대화**)

Task
Natural language inference
Question Answering
Sentence similarity
Classification

Task-agnostic model

$$P(y|x_1,\cdots,x_m) = \text{softmax}(h_l^m W_y)$$

$$L_2(C) = \sum_{(x,y)} \log P(y|x_1,\cdots,x_m)$$

Auxiliary Objectives

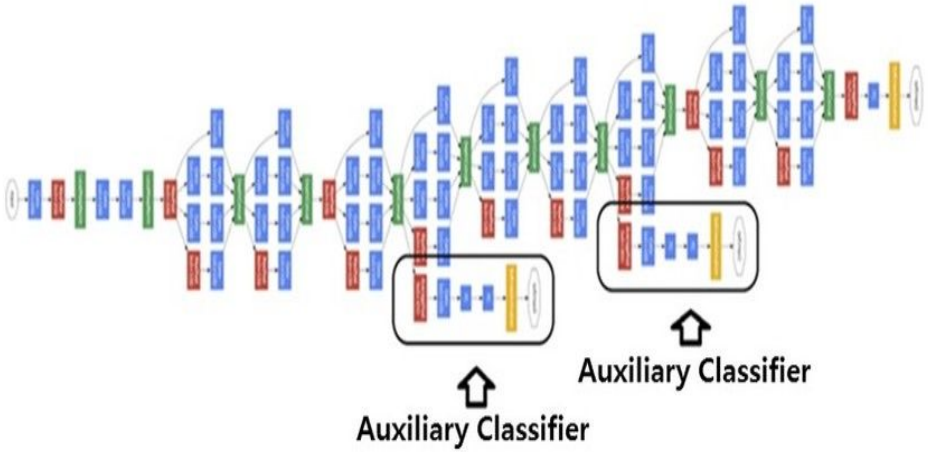
unsupervised pre-training으로 구한 L1사용해서,
L2(C) + λ * L1(C) = L3(C)을 구한다.

$$L_3(C) = L_2(C) + \lambda * L_1(C)$$

이때, **기존의 L1**은 **unlabeled dataset U**에 대한 **L1(U)**로 계산 되었지만, **auxiliary objectives**로서, L1은 **labeled dataset C**에 대해 **L1(C)**로 계산됩니다.

λ는 **L1(C)의 반영 정도**를 정하기 위한 **weight hyper parameter**이다.

이 방법은 **가중치의 수렴을 도우며, 지도학습의 일반화 모델을 향상**시킨다.



Experiments

Task	Datasets
Natural language inference	SNLI [5], MultiNLI [66], Question NLI [64], RTE [4], SciTail [25]
Question Answering	RACE [30], Story Cloze [40]
Sentence similarity	MSR Paraphrase Corpus [14], Quora Question Pairs [9], STS Benchmark [6]
Classification	Stanford Sentiment Treebank-2 [54], CoLA [65]

논문에서는 **각 목적**에 맞게 Labeled dataset을 사용하여, 이전의 모델과 비교를 했습니다. 12개 부문 중 9개 부문에서 **SOTA**를 **달성!!**

✓Natural Language Inference

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

✓Question & Answering

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	77.6	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

✓Semantic Similarity & Classification

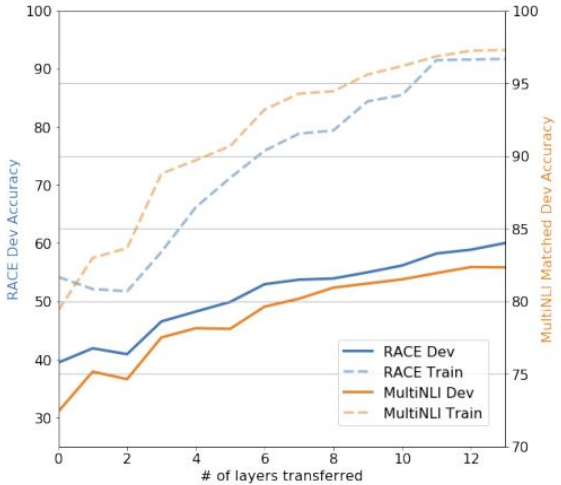
Method	**Classification**		**Semantic Similarity**			**GLUE**
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	68.9
Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

5. 추가적인 실험
[Analysis]

- ❏ Pre-Training의 층이 많을수록 효과적일까?
- ❏ Zero-shot Behaviors

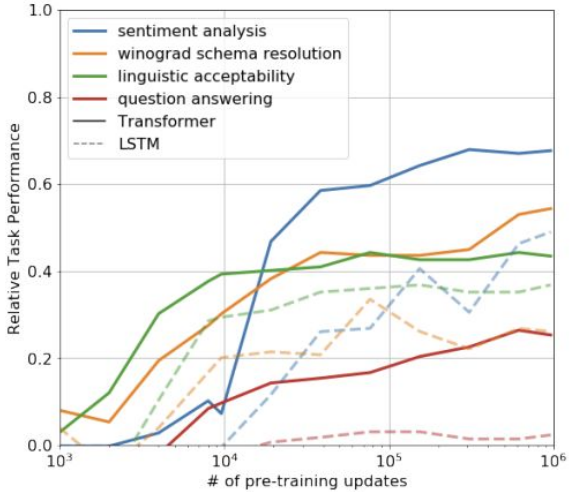
Analysis

1. Pre-Training의 층이 많을수록 효과적일까?



Unsupervised pre-training에서 **transformer layer의 수**에 따른 결과비교

2. Zero-shot Behaviors



Transformer유무와 pre-training에 따른 각각 task의 Zero-shot성능 비교

실선 - Transformer, 점선 - LSTM

대부분의 task에서 **Transformer을 사용할때, 성능이 더 좋음.**
pre-training을 거듭할수록, 그 성능 차이 큼.

특히 **Zero-shot에 대한 LSTM의 성능이 분산이 더 컸다는 결과 :**
pre-training이 전반적인 일반화 능력을 제공한다는 사실을 알 수 있음

GPT-1의 한계와 의의

의의 : RNN같은 구조를 탈피해서 **Transformer 구조를 사용했다는 것**에 의의가 있다.

한계 :

- **Unsupervised learning을 지향했음에도 특정 task에 적용할때, 성능향상을 위해서 fine-tuning과정과 input transformation이 들어갔다는 것**이다.
- 결국 **Fine-tuning과정에서 Supervised learning이 필요로 하는 한계**가 있다.

한계를 해결했을때, 좋아지는 점 :

- 비지도학습만으로 **모델이 만들어지면 더욱 다양하고 범용적으로 사용**할 수 있을 것이다.

GPT-2

변화1(구조)

layer nomalization 위치가 변경, residual layer의 누적에 따라 initalization이 변경, 대용량 데이터를 사용해서 만들고 배치 사이즈도 늘림.

변화2

web scraping dataset으로 다양한 도메인을 이용한 데이터를 사용. fine-tuning없이 unsupervised pre-training만을 통해 General language model 개발

한계

zero-shot, few-shot learning에 대해서는 좋지않은 성능을 보임.특정한 task수행하려면 task-specific datasets, fine-tuning 필요함. 하지만 새로운 task에 충분한 label이 없음.

GPT-3

구조적 특징

1. GPT-3는 GPT-2의 구조에서 모델의 크기, 데이터셋의 크기, 다양성, 학습 횟수를 전반적으로 늘린 모델에 불과하다.

   (BERT는 3억 개, GPT-1은 1억 개, GPT-2는 14억 개 그러나 GPT-3에서는 무려 **1750억 개의 파라미터**를 사용했다.)

2. 하나 다른 점은 full self-attention이 아닌 **sparse self-attention**을 사용한 점이다.

3. 방대한 양의 데이터를 학습하기에 굳이 **fine-tuning 과정을 거치지 않아도 된다.**

   (fine-tuning한 모델을 2021.12에 공개했지만 기존의 GPT-3보다 2배 정도 더 비싸다.)

한계

1. 학습에 필요한 **비용과 시간이 너무 많이 든다.** (한 번 학습하는 데 50~100억 정도 든다고 한다.)

2. 현실세계의 **물리적 상식을 잘 모른다.**

3. **학습에 사용된 예제를 외우고 패턴을 분석, 학습하는 것이지 실제로 추론해내는 것이 아니다.**

4. 제대로 된 **'기억력'이 없다.** (자아를 지닌 인격 모델이 아닌 단순 문장 출력기로 이해하는 것이 바람직하다.)