# Improving Language Understanding by Generative Pre-Training



## Abstract

unlabeled text data는 아주 많지만, 특정 task에 대한 labeled data는 부족하다.

이러한 unlabeled data들을 버리지 말고 generative pre-training하고, 특정 task에 대한 labeled data로 discriminative fine-tuning을 진행하였더니 좋은 성과를 확인할 수 있었다.

---

# Intoduction

NLP 분야에서 labeled data는 매우 부족하기 때문에 되도록 unlabeled data를 잘 활용하여 supervised learning에 대한 의존성을 완화해야 한다.

지도학습이 가능한 경우에도(labeled data가 많은 경우) 미리 unlabeled data로 좋은 representation을 학습해두면 큰 성능 향상이 있다.

하지만, unlabeled data에서 word-level 수준의 정보밖에 얻지 못했고 그 이상의 정보를 활용하는 것은 다음 2가지 이유로 어려웠다.

1. pretrained model을 학습하는데에 어떤 optimization objective가 효과적인지 불분명하다
2. pretrained model을 fine-tuning하는데에 가장 효과적인 방법에 대한 합의가 없다

⇒ unsupervised pre-training과 supervised fine-tuning의 조합으로 language understanding task에 대한 semi-supervised(준지도) 접근 방식을 사용

목표 : 모든 task에 보편적인 representation을 학습하여 이에 대한 최소한의 변화로 task-specific하게 학습하는 것

1. unlabeled data를 활용하여 neural network model의 초기 파라미터를 학습
2. labeled data를 활용하여 목표 task에 맞게 (1)의 파라미터들을 조정

network는 Transformer모델을 사용하였다

- RNN보다 long-term dependency를 핸들링
- task-specific input adaptations을 transfer 시 사용

---

# Related Works

## Semi-supervised learning for NLP

지난 몇년간  unlabeled data에 대해 훈련된 word embeddings를 사용하여 여러 task의 성능을 향상시키는 것을 입증하였지만, 이는 주로 word-level 수준 정보이고, 본 연구에서는 더 높은 수준의 의미를 포착해내고자 했다.

최근 연구에서는 phrase-level이나 sentence-level embeddings를 시도하고 있다.

## Unsupervised pre-training

unsupervised pre-training은 semi-supervised learning의 특별한 케이스로, supervised learning의 좋은 initialization point를 찾는 것이 목적이다.

최근 pre-training은 이미지 분류, 음성 인식, 기계 번역 등 다양한 분야서 도움이 되긴 하지만, 기존에는 보통  LSTM을 사용했기에 긴 문장을 수용할 수 있는 능력을 가지지 못했다.

따라서 본 논문은 Transformer를 사용하여 긴 문장을 수용할 수 있도록 하였다.

## Auxiliary training objectives

unsupervised pre-training의 목적함수를 supervised fine-tuning할 때 보조 학습 목적함수를 추가해주었다.

본 논문에서 보조 학습 목적함수를 추가하였지만, unsupervised pre-training에서 이미 target task와 연관된 많은 언어적 측면을 학습한다.

---

# Framework

## Unsupervised pre-training

### standard LM Objective

unsupervised 말뭉치 토큰 U = $\{u_1, \ldots, u_n\}$이 주어질 때, 아래 확률을 최대화한다.

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

- $u_{i-k}, \ldots, u_{i-1}$가 주어졌을 때, $u_i$를 예측할 확률을 계산
  - ex) I loye you문장에서 I, love가 주어졌을 때 you를 예측하는 확률값
  - k : context window size
  - $\theta$ : SGD로 학습

### GPT-1 Objective

본 논문에서는 Transformer의 decoder부분만을 사용하였다.

- $h_0$ : token embedding($W_e$) 후 position embedding($W_p$)과 더하기

$$h_0 = UW_e + W_p$$

  - U = token의 context vector
- $h_l$ : layer 개수인 n만큼 decoder block 통과하며 학습을 진행 (n=12)

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

- $P(u)$ : position-wise layer($W_e^T$)를 거쳐 softmax로 확률값 구하기

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

→ unlabeled data를 Transformer decoder에 학습시켜 LM(Language model)로 사전 학습을 한 것

## Supervised fine-tuning

### Supervised Target Task Objective

labeled dataset C가 있다고 가정하자. C는 $\{x^1, ..., x^m\}$로 이루어진 input token sequence와 label $y$로 구성되어 있다.

- pretrain된 모델의 position-wise layer와 softmax layer 사이에 linear layer($W_y$)를 추가하여 각 task마다 layer $y$를 예측

$$P(y|x^1, ..., x^m) = softmax(h_l^m W_y)$$

- $x^1, ..., x^m$가 주어졌을 때, $y$를 예측할 확률을 계산

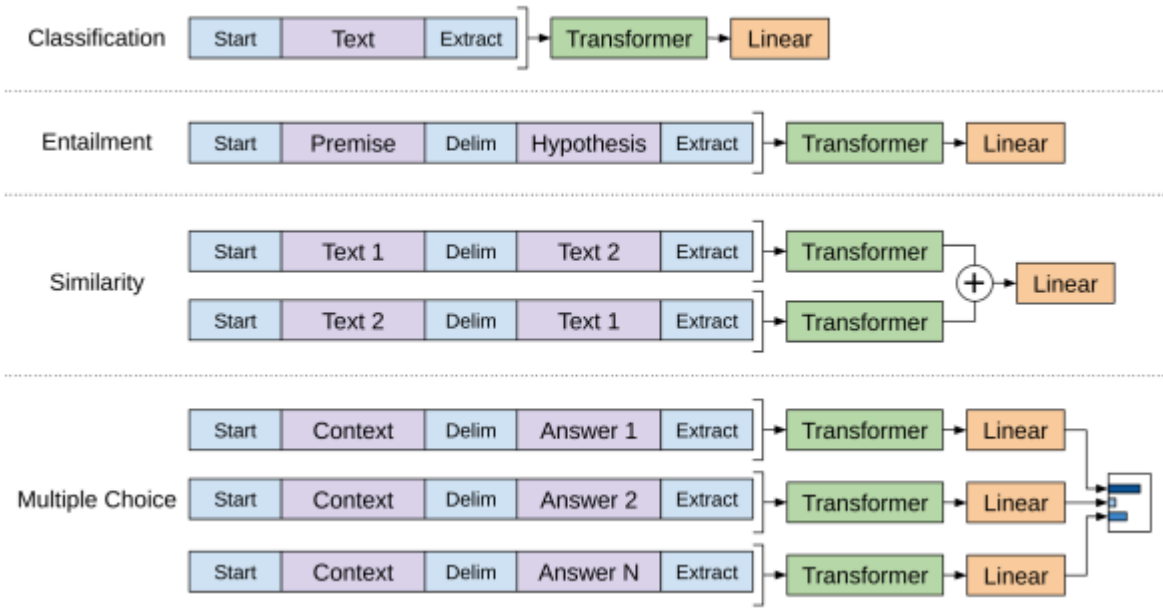
$$L_2(C) = \sum_{(x,y)} logP(y|x^1, ..., x^m)$$

### Auxiliary Objective

- supervised model의 일반화 능력을 향상시키고, 수렴을 가속화하기 위해 pretrained model 기반의 loss($L_1$)과 fine-tuning model 기반 loss($L_2$)를 가중합한 loss를 사용

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

## Task-specific input transformation




### Classification

기존 분류 문제와 동일

### Textual entailment

전제(premise)와 가설(hypothesis), 이 두가지의 sequence token들을 구분하기 위해 기호 $를 사용하여 연결하여 한번에 network에 forward한다

### Similarity

두 문장의 유사성을 비교할 때에는 두 문장 간의 순서가 없기 때문에, {(text1),(text2)}, {(text2),(text1)} 이 2가지를 각각 모델에 forward하여 linear output layer를 거치기 전에 element-wise로 합하여 출력한다

### Question Answering and Commonsense Reasoning

지문 z, question q, 정답 set$\{a_1, a_2, \ldots, a_k\}$를 받아 각 정답 set에 있는 k만큼을 각각 독립적으로 모델에 forward하여 각 softmax를 구해 가장 정답에 가까운 값을 구한다

---

# Experiments

## Unsupervised  pre-training

Book Corpus dataset을 이용하여 LM(Language Model)을 훈련시켰다.

- 7000개가 넘는 다양한 장르의 출간되지 않는 책들로, 길이가 긴 text가 포함되어 있어 생성 모델이  long-range 정보를 학습하기에 좋다

### Model specifications

- transformer decoder layer : 총 12층
- self-attention head : 각 64개의 Q, K, V와 총 12개의 heads로 구성
- position-wise feed-forward : 총 3072차원
- Adam optimizer 사용

## Supervised fine-tuning

### NLI(Natural Language Inference)

: 한 쌍의 문장이 비슷한 내용인지, 연관이 없는 내용인지, 반대되는 내용인지를 추론하는 task

| 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 |

- 5개 dataset 중 4개에 대해 이전 SOTA모델을 개선

## QA and Commonsense reasoning

| 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** |

- RACE dataset(중고등학교 시험문제와 관련된 영어 지문)을 사용했으며, long-range context를 효과적으로 처리하는 능력을 보여주었다.

## Semantic Similarity and 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** |

# Conclusion

GPT-1은 task별로 architecture를 설계하는 것이 아닌, generative pre-training 과 discriminative fine-tuning을 제안하였다.

이를 통해 question answering, semantic similarity  assessment, entailment determination, text classification 등 12개 중 9개 task에서 state of the art를 달성했다.