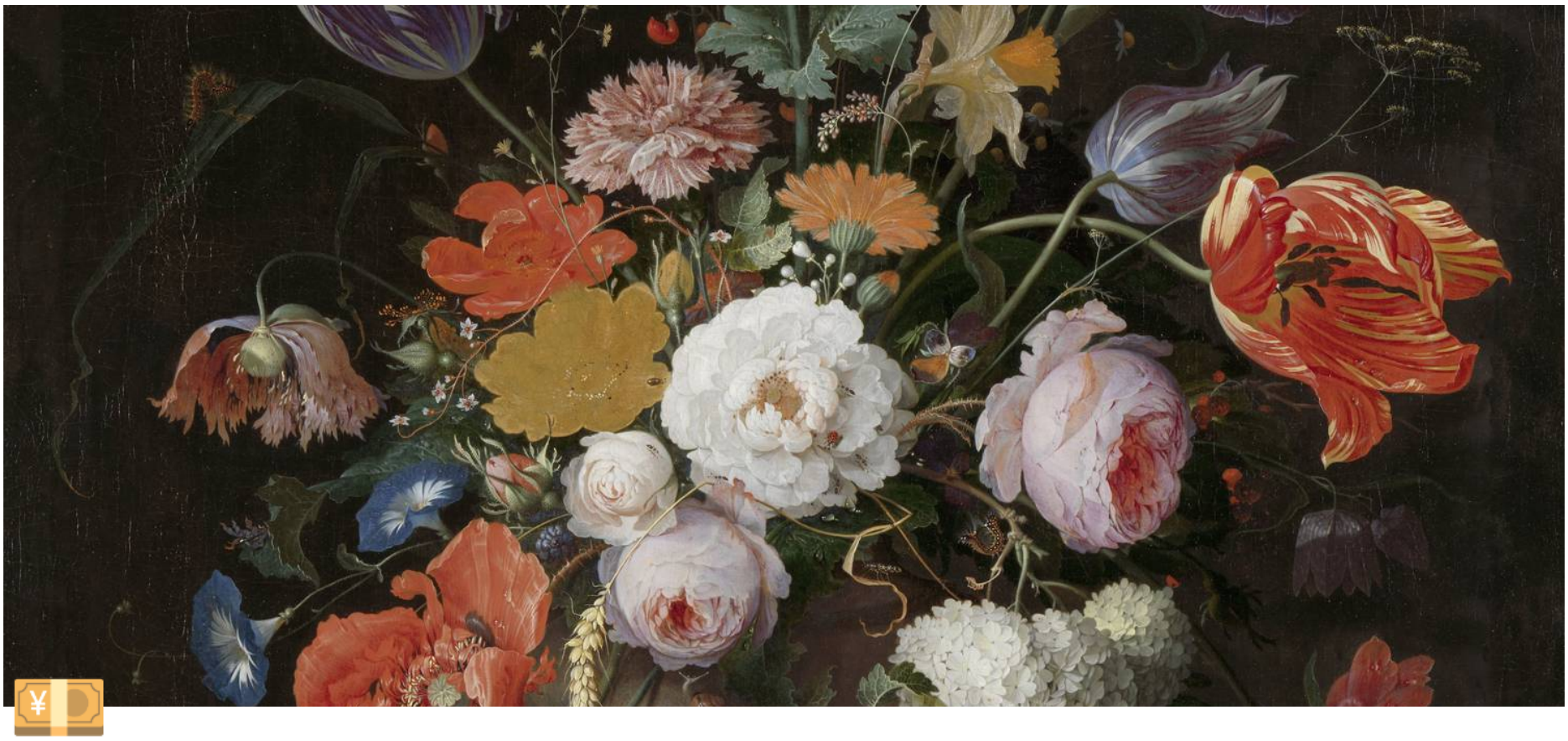

GPT-1

Improving Language Understanding by Generative Pre-Training

Abstract

Natural language이해는 textual entailment, question answering, semantic similarity assessment, and document classification와 같은 다양한 작업으로 구성됩니다. 텍스트 데이터는 풍부하지만 특정 작업을 위한 라벨은 부족해서 모델을 적절히 훈련하는 것은 어렵습니다. 우리는 언어 모델의 generative pre-training과 각 테스크에 맞춘 discriminative fine-tuning에 의해 위와 같은 테스크에 많은 발전을 가져다 줌을 보입니다. 이전과는 다른 접근으로 모델 구조의 최소한의 변경을 통해 각 테스크에 맞춘 좋은 성능을 나태내기 위한 fine-tunning을 진행합니다. 이를 통해서 12개의 task중 9개에서 sota를 달성하게 된다.

Introduction

원본 텍스트로부터 효과적으로 학습하는 방법은 NLP에서 supervised learning의 의존성을 완화시키는 것이다. 대부분의 딥러닝 방식은 대게 라벨 데이터를 필요로 하지만 이는 라벨 데이터가 부족하여 많은 테스크에서 제약이 생깁니다. 이러한 상황에서 시간이 많이 들며 비싸지만 언라벨 데이터에서 언어 정보를 활용할 수 있는 모델은 annotation을 모으기 위해 유용한 대안을 제공합니다. 이러한 경우 supervision 학습이 가능하지만, unsupervised에서 좋은 representation은 상당한 성능 향상을 가져다 줄 것입니다. 위에 대한 근거로는 NLP테스크에서 광범위한 pre-trained word embedding을 사용할 경우 성능이 향상했다는 것입니다.

unlabeled text에서 더 많은 word-level 정보를 활용하는 것은 두가지의 이유로 인해 문제가 있습니다. 먼저 어떤 목적 함수가 특정 task에 대한 representation을 잘 만들어낼지 불분명합니다. 최근 연구에서 다양한 목적함수가 사용된 task에서 좋은 성능을 보였습니다. 두번째로, 학습된 representation을 특정 task로 맞추는 것에 대한 효과적인 방법이 없습니다. 현재 방법에는 특정 테스크에 따른 모델 구조 변경, 복잡한 학습 방식 그리고 보조 목적 함수가 있습니다. 이러한 애매한 것들은 언어 모델에 대한 효과적인 semi-supervised learning을 발전시키기 어렵게 합니다.

이 논문에서는 언어 테스크에 대해 unsupervised pre-traing 과 supervised fine-tunning을 활용한 semi-supervised 방식을 제안합니다. 우리의 목표는 작은 적용을 통한 범용적인 representation을 학습하는 것입니다. 우리는 많은 unlabeled text 말뭉치 데이터 세트와 annotated가 지정된 데이터셋이 있다고 가정합니다. 그리고 target task가 unlabeled 말뭉치와 동일한 도메인이 아니어도 됩니다. 우리는 두가지의 학습 방식을 이용하는데 먼저, 언어 모델 목적함수를 unlabeled 데이터가 새로운 언어 모델 초기 가중치를 학습하는데 사용합니다. 연속적으로 이러한 가중치들은 supervised objective와 연관되어 target task에 사용됩니다.

우리는 다양한 기계 번역, 문서 생성, 구문 분석과 같은 다양한 테스크에서 좋은 성능을 나타내는 transformer구조를 사용합니다. transforemr는 구조화된 메모리를 통해 text에서 long-term dependencies를 해결합니다. 전이학습에 경우, 특정 테스크에 인풋에 따른 스타일을 활용합니다. 이는 fine-tune을 최소한의 pretained model의 변화로 효과적으로 가능게 합니다.

우리는 natural language inference, question answering, semantic similarity, and text classification를 통해 평가를 진행합니다. 우리의 방식은 특정 테스크에 맞춰 학습하는 방식보다 더 좋은 성능을 발휘했습니다.

또한, pre-trained model에서의 zero-shot을 분석해본 결과 pre-trained model은 다른 테스크의 유용한 언어적 지식을 얻는다는 것을 증명했습니다.

Related Work

Semi-supervised learning for NLP

우리의 작업은 semi-supervised learning of NLP에 속합니다. 이러한 관점은 sequence labeling과 text classification에 지대한 관심을 이끌었습니다. 초기 연구에서는 unlabel 데이터를 work-leveel 이나 phrase-level의 통계를 계산하는데 사용되어 upervised model의 feature로 사용되었습니다. 지난 몇년간 연구자들은 unlabel 말뭉치로 학습되어진 단어 임베딩 방식의 장점을 증명하였지만, 이런 방식은 단어 정보를 파악하는데 그치며 우리는 더 높은 단계의 의미를 포착하고자 합니다.

최근 연구는 unlabel 데이터에서 단어 수준의 의미보다 더 높은 차원의 정보를 활용하고자 하고 있습니다. 문맥 단계나 문장 단계 임베딩은 unlabel 데이터를 사용하여 학습할 수 있으며 다양한 테스크에서 적절한 벡터 표현으로 인코딩됩니다.

Unsupervised pre-training

unsupervised pre-training은 좋은 파라미터 초기 지점을 찾는 semi-supervised learning의 특이한 경우입니다. 초기에는 이미지 분류나 회귀에서 사용되었습니다. 이어진 연구에서는 pre-training이 딥러닝에서 정규화 역할을 해주는 것을 밝혔습니다. 최근에는, 다양한 연구에서 활용되고 있습니다.

우리와 가장 유사한 작업은 pre-training network를 언어 모델을 사용하여 학습시킨 후 supervised 학습을 통해 fine-tunning을 시키는 것입니다. 그러나 이들은 LSTM models을 사용하였고 이는 짧은 길이라는 제약이 있습니다. 우리는 transformer를 사용하여 더 긴 길이를 사용할 수 있습니다.

또 다른 접근은 pre-trained model에서 나온 hidden representations를 보조 피쳐로서 타겟 테스크에 맞춘 지도 학습에 사용했습니다. 이는 테스크에 맞춰 많은 양의 새로운 파라미터가 필요하지만 우리는 테스크에 따라 약간의 변경만을 요구합니다.

Auxiliary training objectives

보조 목적 함수를 비지도 학습에 추가하는 것은 반지도 학습학습에 방식입니다. 초기에는 다양한 보조 테스크를 의미적 라벨링을 부여하기 위해 사용했다. 최근에는 보조 언어 목적 함수를 지도 목적 함수에 추가했고 이는 더 좋은 성능을 나타냄을 보였습니다. GPT 또한 이를 사용하지만, 비지도 학습은 이미 타겟 테스크와 관련되어 있는 언어적 측면을 학습합니다.

Framework

학습 과정은 two-stage로 이루어져 있다. 먼저 정말 많은 텍스트를 범용적인 언어 모델을 학습하는 것이다. 그 후, 라벨 데이터를 통해 테스크에 맞춘 모델을 적용시켜 fine-tunning을 진행합니다.

Unsupervised pre-training

주어진 학습되어지지 않은 단어 $U = [u1, ..., u2]$를 표준적인 언어 모델링을 통해 likelihood를 극대화 시킵니다.

$$L1(U) = \Sigma_i logP(ui|ui-k, ..., ui-1; \Theta)$$

k는 context window의 크기이며 조건부 확률 P는 파라미터 Θ를 갖는 신경망으로 모델링됩니다. 그리고 stochastic gradient descent를 통해 학습이 이루어집니다.

우리는 멀티 레이어 트랜스포머 디코더를 기존의 트랜스포머에서 약간의 변형을 통해 사용합니다. 이 모델은 multi-head self-attention기반입니다.

$$\begin{aligned} h_0 &= UW_e + W_p \\ h_l &= \texttt{transformer\_block}(h_{l-1}) \forall i \in [1, n] \\ P(u) &= \texttt{softmax}(h_n W_e^T) \end{aligned}$$

Supervised fine-tunning

위에서 비지도 pre-training을 진행한 다음 우리는 타겟 테스크에 대해 지도학습을 진행합니다. label dataset C를 sequence input tokens $x1, ..., x^m$이 label y로 가정하자. 인풋은 pre-trained model을 통과하며 마지막 트랜스포머의 activation block을 거쳐 $P(y|x^1, ..., x^m) = softmax(hl^m Wy)$.으로 나온다. 그리고 이는 loglikelihood를 최대화 시킨다. $L_2(C) = \Sigma(x, y) logP(y|x^1, ..., x^m)$. 게다가 fine-tinong에 보조 목적함수로 언어 모델을 추가하는 것은 지도학습에 정규화를 증진시켜줌과 융합을 가속화시킵니다.

$L3(C) = L2(C) + \lambda * L1(C)$. 전체적으로 fine-tunning간 추가로 요구하는 파라미터는 fine-tinning간 Wy와 구분 토큰이다. ex)sos, eos

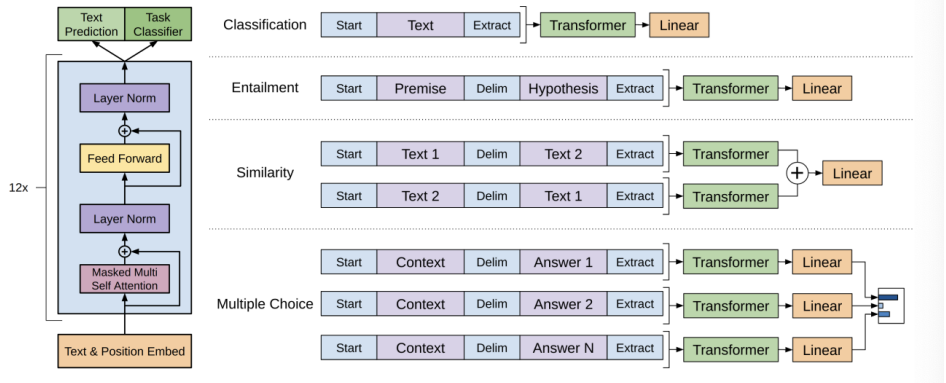

Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Task-specific input transformations

텍스트 구분같은 몇몇 테스크에서 우리는 직접 모델을 fine-tunning이 가능합니다. question answering, textual entailment과 같은 다른 테스크에서는 우리의 pretrained model은 연속적인 텍스트를 통해 학습되어졌기 때문에 약간의 수정이 필요합니다. 이와 같은 접근은 많은 양의 테스크 별 수정을 다시금 요하며 이는 전이학습에서 추가적인 모델 변경을 필요로 하지 않습니다. 대신에 우리는 pretrained 모델이 진행시킬 수 있도록 구조화된 Input을 sequence 형식으로 변환합니다. 이러한 변환된 인풋은 테스크 별 많은 수정을 회피할 수 있습니다. 우리는 아래에서 이러한 변형 방식을 간결히 설명합니다. 모든 변형은 랜덤적으로 sos, eos 토큰을 추가합니다.

Textual entailment

premise p와 hypothesis h token 사이에 delimiter token ($)를 사이에 두고 이어 붙여 sequence를 구성합니다.

Similarity

delimiter token ($)를 사이에 두고 앞문장 뒷문장 그리고 뒷문장 앞문장해서 독립적인 token sequences를 구성합니다.

Question Answering and Commonsense Reasoning

document z, question q 그리고 possible answers $[a_k]$가 주어진다면 z, q를 합쳐 context sequences를 생성하고 delimiter token을 추가하고 가능한 답변을 쭉 나열하여 여러 독립적인 sequences를 구성합니다.

Experiments

Setup

- GPT는 transfomer의 12개의 decoder layer를 사용하여 masked self-attention heads로 768차원을 가진다.
- position-wise feed-forward에서는 3072 hidden dimenstion으로 사용한다.
- Adam optimization을 max learning rate 2.5e-4.를 사용한다. LR은 200번 업데이트할때 까지는 증가하며 다시 cosine scedule을 통해 0까지 줄어든다.
- 레이어 노말라이제이션이 전반적으로 사용되어 초기 가중치 초기화는 N(0, 0.02)는 충분하다.
- 그리고 40000개의 병합이 된 BPE인코딩을 사용하며 residual, embedding, attention drop out은 0.1로 설정한다.
- 수정된 L2 regularization을 사용한다.
- Gaussian Error Linear Unit을 활성화 함수로 사용한다.
- 기존과 다르게 position embeddings는 학습하여 진행한다.
- ftfy library를 통해서 문장 부호 및 여백을 제거하며 spaCy tokenizer를 사용한다.

Supercised fine-tunning

`Natural Language Inference`

NLI는 textual entailment로 알려져 있으며 두 문장간 관계를 파악하는 것이다. 이 테스크에서는 lexical entailment, coreference, lexical … 등 다양한 현상으로 인해 어려움을 겪고 있다. 우리는 5가지 데이터 셋에 대해서 평가를 진행한다. 아래에 표에서 대부분의 데이터셋에서 성능 향상을 이뤄낸 것을 확인할 수 있다.

Table 2: Experimental results on natural language inference tasks, comparing our model with current state-of-the-art methods. 5x indicates an ensemble of 5 models. All datasets use accuracy as the evaluation metric.

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 and commonsense reasoning`

우리는 중,고등학교 시험에서 추출한 RACE dataset을 사용합니다. 이 데이터셋은 보다 많은 추론 문제가 존재했습니다. 두 가지의 답변 옵션 중 하나를 선택하는 과제이며 우리의 모델은 많은 성능 향상을 이뤄냈습니다.

Table 3: Results on question answering and commonsense reasoning, comparing our model with current state-of-the-art methods.. 9x means an ensemble of 9 models.

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`

두 문장간의 의미적 유사도를 추론하는 테스크이다. 개념의 재구성과, 부정의 이해와 의미적 모호함을 다루는데 문제가 놓여있다. 우리는 세가지의 데이터 세으로 평가를 했고 두개의 데이터 셋에서 sota를 달성한다.

`Classification`

우리는 두가지 classification을 진행한다. 하나는 문장이 문법적으로 옳은지에 대해서와 하나는 binary classfication task이다. 둘 다 많은 성능 향상을 이뤄냈다.

Table 4: Semantic similarity and classification results, comparing our model with current state-of-the-art methods. All task evaluations in this table were done using the GLUE benchmark. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

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**

전반적으로 GPT는 12개의 데이터 셋 중 9개에서 sota를 달성하게 된다.

Analysis

Impact of number of layers transferred

비지도 학습 pretrained이 target task에 얼마나 영향을 주는지 확인하기 위해서 pretraining network의 layer 수와 학습 업데이트 정도를 분석해본다. 결과적으로 layer가 많을수록, 학습을 많이 할수록 target task의 결과가 좋아지는 것을 확인할 수 있었다.

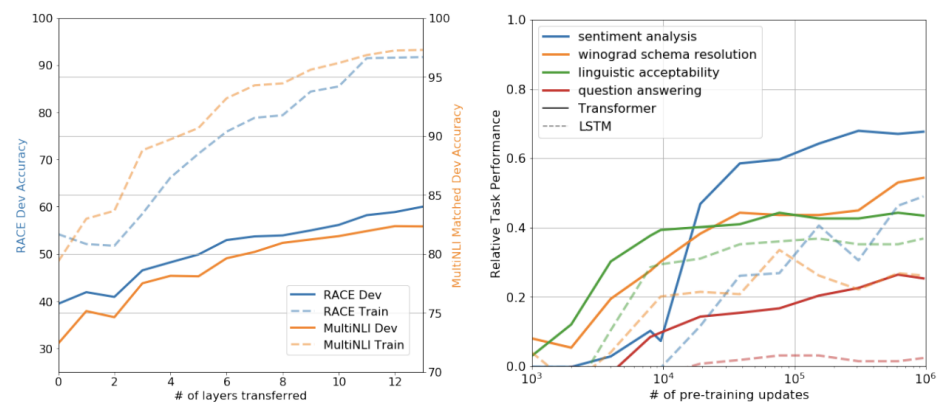

Figure 2: (**left**) Effect of transferring increasing number of layers from the pre-trained language model on RACE and MultiNLI. (**right**) Plot showing the evolution of zero-shot performance on different tasks as a function of LM pre-training updates. Performance per task is normalized between a random guess baseline and the current state-of-the-art with a single model.

Zero-shot Behaviors

어째서 pretraining transformer가 효과적인지 이해해보고자 한다. 가정은 기본적인 생성 모델은 많은 테스크에서 수행하기 위해 배우고 transforemr의 attention mechanism은 LSTM보다 전이 학습에서 더욱 구조적이다라는 것이다. 우리는 지도 finetunning을 제외하고 수행하도록 모델을 구축한다. 우리는 pretrained가 없을 경우 성능이 많이 떨어지는 것을 확인할 수 있었으며 LSTM은 task별 성능 편차가 많은 것을 확인할 수 있었다.

Ablation studies

우리는 세가지 연구에 대해서 진행을 하였다. LSTM과 Transformer의 비교, 보조 목적함수의 유무, pre-training의 유무를 통해 결과를 비교한다.

Table 5: Analysis of various model ablations on different tasks. Avg. score is a unweighted average of all the results. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

Method	Avg. Score	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	MNLI (acc)	QNLI (acc)	RTE (acc)
Transformer w/ aux LM (full)	74.7	45.4	91.3	82.3	82.0	**70.3**	**81.8**	**88.1**	**56.0**
Transformer w/o pre-training	59.9	18.9	84.0	79.4	30.9	65.5	75.7	71.2	53.8
Transformer w/o aux LM	**75.0**	**47.9**	**92.0**	**84.9**	**83.2**	69.8	81.1	86.9	54.4
LSTM w/ aux LM	69.1	30.3	90.5	83.2	71.8	68.1	73.7	81.1	54.6

Conclusion

우리는 pre-training괴 fine-tunning을 통한 강력한 언어 모델을 제시한다. pre-training을 통해서 다양한 단어뭉치에서 좋은 성능을 얻을 수 있었으며 전이 학습을 성공적으로 진행시켰다. 그리고 이는 12가지의 데이터 셋 중 9개에서 소타를 달성한다.