Transformer-XL

Long-Term dependency를 요구하는 Language Modeling(향후 글에서는 축약어인 LM으로 대체체합니다.)은 수많은 unsupervised pretraining의 목적함수로 사용되었다. 초반에는 RNN으로 구현되었던 LM은 long-term dependency를 해결하기 위한 Gate의 도입(LSTM,GRU), 원타임으로 멀리 있는 token에 접근가능한 Attention의 도입으로 성능향상을 이어왔다. 하지만 Transformer기반 LM 모델도 결국 Attention에 의한 연산량 증가로 512나 1024 Token까지 밖에 받지 못하는 한계가 존재한다.

결국 문장의 맥락적 의미를 고려하지 않고 단지 Max\_Length에 맞춰서 Token을 잘라서 훈련할 수 밖에 없는 환경은 LM의 진정한 능력인 Long-Term dependency학습에 한계가 있다.  
(ex. 각 Segment길이가 512라고 할 때, [[Segment01], [Segment02], [Segment03]]로 이루어진 문서에서 기존의 LM이 Segment02를 학습할 때 Segment01의 이전 맥락을 전혀 활용하지 못한다.)

이러한 문제들을 하기 위해서 저자들은 아래의 2가지 테크닉이 추가된 Transformer-XL (XL=Extra Large)을 제시한다.

**1. Segment-Level Recurrence with State Reuse**

말그대로 이전 Segment(t-1)의 Hidden State를 활용해 현재 Segment(t)에서의 LM을 수행하는 방법을 말한다. 본 논문과 추후 XL-NET에서 저자들은 이전 Segment를 'Memory'라고 명명한다.

**2. Relative Postional Encoding (PE)**

이전 Segment들(t-1, t-2, ...)을 활용하면, PE(Positional Encoding을 앞으로 PE로 대체합니다) 설계에 있어서 문제가 생겨버립니다. (ex. Segment02의 PE는 Segment01의 PE와 다라야 한다) 따라서 저자들은 기존의 Word Embedding에 PE를 더하는 방식이 아닌 (**PE의 본질 : Position을 고려하는 이유는 0,1,2와 같은 순차적인 정보보다 토큰 간의 거리 정보를 넣어주기 위함**), Attention 연산 시에 Token간의 상대적인 위치를 반영하도록 하는 새로운 방법론을 제시한다.

XL Net

XLNet은 최근에 대부분의 NLP 테스크들에서 state-of-the-art 성능을 달성하고 있던 BERT를 큰 차이로 outperform 하면서 파장을 일으켰습니다. 논문 저자들은 기존의 Language Model들과 BERT의 문제를 지적하면서 새로운 학습 방식과 이 학습 방식을 적용하기 위한 모델을 제안합니다. 이 논문의 핵심 contribution은 다음과 같습니다.

* XLNet은 GPT로 대표되는 auto-regressive(AR) 모델과 BERT로 대표되는 auto-encoder(AE) 모델의 장점만을 합한 generalized AR pretraining model입니다.
* 이를 위해 permutation language modeling objective과 two-stream attention mechanism을 제안합니다.
* 다양한 NLP 테스크에서 기존 대비 상당한 향상을 보이며 state-of-the-art 성능을 보였습니다.

AR과 달리 AE는 특정 [MASK] token을 맞추기 위해 양 방향의 정보를(Bi-Directional Self-Attention) 이용할 수 있다는 장점이 있습니다. 하지만 independant assumption으로 모든 [MASK] token이 독립적으로 예측됨으로써 이들사이의 dependency를 학습할 수 없다는 치명적인 단점이 있습니다. 또한 noise([MASK] token) 자체는 실제 fine-tuning 과정에는 등장하지 않으므로, pre-training 과 fine-tuning 사이의 불일치가 발생합니다.

* <https://ratsgo.github.io/natural%20language%20processing/2019/09/11/xlnet/>