**GPT와 같은 Transformer Architecture를 Multimodal로 확장하는 방법**

**서강대 AI SW 대학원 A70071 유지연**

Multimodal로 확장하는 방법을 기술하기에 앞서, 아키텍처를 이해하는 것이 중요하다고 생각되어 간략히 Transformer 모델 구조에 대해 기술하고, BERT 와 GPT에서 Transformer 가 어떻게 다르게 쓰이는지 기술하고, 최종적으로 GPT 와 같은 Transformer 구조를 멀티모달로 확장하는 방법에 대해 기술한다.

**1. Transformer Architecture 개요**

Transformer는 encoder-decoder 아키텍처에서 흔히 사용되는 recurrent layer를 multi head self-attention 으로 대체하여, 전적으로 Attention mechanism 에 기반하도록 설계된 최초의 Sequence transduction (시퀀스 변환) 모델이다.[[1]](#endnote-1)

Transformer의 encoder-decoder 구조를 설명하기에 앞서, encoder와 decoder의 기본 개념을 정리하면 아래와 같다.

Encoder는 사람이 사용하는 자연어와 같은 데이터를 기계가 이해할 수 있는 숫자 벡터로 변환하는 역할을 한다. 예를 들어 “I am Engineer” 라는 문장이 있으면 각 단어를 벡터로 변환하고 이 벡터들을 조합해서 전체 문장의 정보를 가진 최종 벡터 (인코더의 출력 벡터)를 만드는 것이다.

Decoder는 인코더가 만들었던 문장 벡터를 입력으로 받아서, 사람이 필요로 하는 자연어 단어 한 개를 만들고, 문장 벡터와 이 생성된 자연어 단어를 포함해서 다시 그 다음 자연어 단어를 만들고 반복하는 방식으로 최종적으로 자연어 문장을 만드는 역할을 한다. 이러한 방식을 Auto-Regressive 방식이라고 하는데, 모든 seq2seq 모델의 decoder가 사용하는 방식이다.

그렇다면 Transformer 의 encoder-decoder는 어떻게 다를까?

앞서 설명했던 것처럼, 일반적인 encoder-decoder는 순차적으로 시퀀스를 처리하여 긴 시퀀스의 경우 장기 의존성 문제를 겪는데, Transformer 의 encoder-decoder 는 각 단어가 다른 모든 단어와의 관계를 학습하는 Self attention 구조를 갖기 때문에 병렬 처리가 가능하며, 장기 의존성 문제가 해결된다. 또한 순서를 알지 못하는 일반 self attention 구조를 보완하기 위해 시퀀스의 위치를 담는 positional encoding 방식을 추가로 사용하여 문장의 순서 정보 또한 보존한다.

즉 순차 처리가 아니면서도 순서 정보를 가질 수 있다는 것이 핵심이다.

**2. BERT와 GPT의 Transformer**

‘GPT와 같은 Transformer’ 를 확장시키는 방법을 고민하기에 앞서, 이번엔 분류 모델로 대표적인 BERT 와 생성 모델로 대표적인 GPT의 Transformer 가 어떻게 다르게 사용되는지 알아보자.

BERT는 Bidirection Encoder Representations from Transformers를 뜻한다. 이름에서 알 수 있듯, Transformer의 encoder 를 사용해서 입력 시퀀스를 ‘양방향’으로 학습하는 모델이다. 문장 내에서 단어를 숨기고 숨긴 단어를 예측하도록 하는 masked language modeling 기법[[2]](#endnote-2)과, 문장 내 모든 단어와의 관계를 학습하는 Self-attention 메커니즘을 통해, 단방향 모델과 비교하여 더 많은 정보를 하여 문장을 이해하는 데에 특화되어 있다.

반면 GPT는 Generative Pre-trained Transformers 로, Transformer의 decoder 만을 사용해서 auto regressive 방식으로 학습한다. Auto regressive 란 이전 단어를 기반으로 다음 단어를 순차적으로 예측하는 것을 말한다. Self-attention의 특징이 모든 단어 간의 관계를 학습하는 것인데 여기서 ‘순차적으로 예측’ 한다는 표현에 대해 의문이 들 수 있다. 부연 설명을 하자면, mask 기법을 사용하여 예측해야 할 단어를 기준으로 그 이후의 시퀀스들은 숨김 처리를 하여, 그 이전 단어까지만 참조해서 그 이후의 단어를 예측한다는 점에서 ‘순차적’이고 ‘단방향성’이라는 의미이다. 이전 단어들 사이에서는 모든 단어 간의 관계를 학습하는 ‘self-attention’ 메커니즘이 적용되어 양방향 관계를 학습하고 단어간의 관계를 파악한다. 이 때문에 GPT 모델은 다음 단어를 예측하고 문장을 생성하는 데에 특화되어 있다.

**3. GPT와 같은 Transformer 를 Multimodal로 확장하는 방법**

그러면 GPT와 같은 Transformer 를 multi modal로 확장하려면 어떻게 할까?

먼저, Attention 메커니즘에 대해 더 자세히 설명하면 아래와 같다.

Attention Mechanism 은 Query, Key, Value 간의 관계를 학습한다. Query 는 현재 모델이 집중하려는 대상이며, Key 는 중요한 정보의 인덱스 역할, Value 는 실제 참조할 데이터를 말한다. 여기서 텍스트와 이미지 데이터를 결합하는 multi modal 을 만들기 위한 힌트를 얻을 수 있다.  
 예를 들어, <장화신은 고양이 사진>이 있고, 해당 사진을 설명하는 캡션 텍스트 “고양이가 장화를 신고 있다.” 가 있다면, ‘고양이’ 텍스트를 인코더에 통과시켜 생성한 벡터를 Query-1, ‘장화’ 텍스트를 인코더에 통과시켜 생성한 벡터를 Query-2 로, 이미지 내에서 고양이 픽셀 부분을 인코더에 통과시켜 생성한 벡터를 Key-1, Value-1 장화 픽셀 부분을 통과시켜 생성한 벡터를 Key-2, Value-2 라고 표현할 수 있을 것이다. 그리고 어텐션 메커니즘에 따라, Query-1이 Key-1 을 참조하고 Query-2가 key-2를 참조하도록 하는 것이다. 사실 이는 개념적으로 예를 드는 것인데, 실제로는 이미지 전체가 Key, Value 이며 특정 영역 (장화가 위치한 부분에)에 Query가 높은 가중치(해당 위치 Query가 ‘장화’ 임에 가중치 부여) 를 부여하는 것이다. 이렇게 하면 전체 이미지와 텍스트를 함께 학습하는 멀티 모달 구조를 얻을 수 있다. 이와 같은 메커니즘을 Cross modal attention 이라고 부른다.

또 어떤 방법이 있을까?

Transformer는 기본적으로 텍스트 시퀀스를 처리하도록 설계되어 있다. 그렇다는 것은 이미지를 텍스트와 같은 벡터 형식으로 변환한다면 모델이 정보를 이해하고 학습할 수 있을 것이다.

이와 관련한 기법을 Vision Transformer 혹은 patch embedding 이라고 한다. 이미지를 작은 패치 단위로 분할해서, 패치들을 텍스트 토큰처럼 처리하는 것인데, 실제로 텍스트를 처리하도록 설계되어 있는 Transformer 를 이용해서 이미지를 학습하도록 확장시킨 Vision Transformer 라는 모델이 있다.  
 해당 모델은 멀티 모달은 아니지만, Transformer 아키텍처를 사용하여 이미지를 텍스트와 같은 방식으로 학습하는 모델로 해당 모델을 통해 multi modal 로 확장하는 아이디어를 얻을 수 있다.  
 Vision Transformer 는 N x N 크기의 패치로 나눈 이미지를 Linear Projection을 통해 특징 벡터를 생성하고, 변환된 해당 특징 벡터를 Transformer 의 입력 토큰으로 사용하여 이미지를 텍스트와 동일한 방식으로 학습을 한다. 이제까지 설명했던 Transformer 아키텍처와 마찬가지로, multi head self attention 을 통해, 각 이미지 패치와 다른 이미지 패치와의 관계를 학습하며, positional embedding 을 통해 이미지 위치 정보 또한 보존하며[[3]](#endnote-3) 학습을 하는 것이다.

Transformer를 이미지로 확장시킨 위 개념을 적용하여 실제로 Multi modal 로 확장시킨 모델로는 CLIP 이 있다.

CLIP은 Contrastive Language Image Pretraining의 약자로, 텍스트는 Transformer 를 기반으로, 이미지는 Vision Transformer 혹은 ResNet을 기반으로 처리해서 학습한 분류형 멀티 모달 모델이다. 텍스트는 GPT 와 마찬가지로 입력 텍스트를 토큰화 하고 Positional encoding 을 추가하여 변환한 벡터를 입력으로 사용한 Transformer 아키텍처를 적용했으며, 이미지는 ViT 혹은 ResNet 등을 사용해 이미지를 패치로 나누고 Linear Projection 을 통해 특징 벡터를 생성해서, 이를 Transformer의 입력으로 사용하였다. 당연히 단순히 각각 Transformer encoder를 통해 벡터화 한다고 해서, 텍스트와 이미지를 동시에 관계를 파악하고 이해할 수는 없다. 때문에 CLIP 은 어떤 이미지-텍스트가 올바른 쌍을 이루고 잘못된 쌍을 이루는지 판단할 수 있는 것을 목표로 하여, Contrastive Learning이라는 대조 학습을 통해, 이미지 encoder 와 텍스트 encoder 를 통해 동일한 벡터 공간으로 임베딩하였다. (동일한 벡터 공간으로 임베딩 한다는 것은, 동일한 수학적 표현으로 나타냄을 뜻한다. 즉 고양이 이미지가 변환된 벡터와 ‘고양이’라는 텍스트가 변환된 벡터가 가까운 위치에 있도록 임베딩 한다는 것이다.) 각각의 인코더를 통과한 벡터를 학습 데이터로 사용하여 같은 의미의 text-image 쌍을 cosine 유사도를 최대화하여 가까운 위치로, 다른 의미의 text-image 쌍을 cosine 유사도를 최소화하여 먼 위치로 가도록 최적화 하며 학습시킨 것이다.[[4]](#endnote-4) 이 모델의 임베딩 학습 방식을 통해 Transformer 아키텍처를 multi modal 로 확장시키는 아이디어를 얻을 수 있다.

하지만 이러한 원리는 GPT와 같은 생성형 모델을 multi modal로 확장하기에는 조금 어려워 보인다. 기본적으로 위와 같은 분류형 모델은 Transformer 의 encoder 를, 생성형 모델은 Transformer 의 decoder를 사용하는 차이가 있기 때문이다. 그렇다면 생성형 모델을 멀티 모달으로 적용하려면 어떻게 해야 할까?

이는 DALL.E 를 통해 배울 수 있다.   
DALLE.E 는 Transformer 기반인 GPT 아키텍처를 사용하는 이미지 생성 모델이다. 앞서,설명했던 것처럼, Transformer의 decoder 를 사용하여 GPT와 동일하게 positional encoding 을 통해 순서 정보를 보존하며 텍스트 데이터를 벡터로 변환한다. 하지만 이미지 처리 방식이 CLIP 과는 다른데, 이미지 데이터의 픽셀 패치를 텍스트와 같은 벡터 공간으로 임베딩 하는 방식이 아닌, Discrete Latent Token 이라고 불리는 토큰 형태로 압축하는 방식을 사용한다. 하지만 결국 원리는 앞서 설명했던 이미지 데이터를 텍스트처럼 처리하기 위한 방식에서 기인한다. 이렇게 변환한 이미지 정보를 가진 토큰을 텍스트와 결합하여 transformer decoder 의 입력으로 사용하며, 마스킹된 self-attention 메커니즘을 통해 text-image 간의 관계를 학습하며 auto regressive 방식으로 다음 토큰을 예측한다. 이렇게 text-image 정보를 가진 예측된 토큰을 다시 원본 이미지로 복원하여 이미지를 생성하는 것이다[[5]](#endnote-5).

정리하면, Transformer 아키텍처의 핵심은 Self-attention 메커니즘과 positioning embedding인데 이는 텍스트 처리에서 처음으로 사용되었다. 때문에 이미지 데이터를 Transformer 아키텍처를 기반으로 학습하고 처리하고자 하는 아이디어로, 텍스트와 마찬가지로 Transformer 가 이해할 수 있는 벡터 형식으로 변환하는 방식이 등장하였다. 이는 Transformer의 encoder를 사용하는 분류 모델에도, decoder 를 사용하는 생성형 모델에도 적용되는 아이디어이다. 하지만 단순히 각각의 데이터를 벡터화 하는 것 만으로는 당연히 두개의 다른 정보를 동시에 이해하는 multi modal 모델을 만들 수 없다. 때문에 분류형 multi modal을 만들기 위해서는 텍스트-이미지 쌍을 이용해 각각의 두 벡터 데이터가 비슷한 의미일 경우 가깝게, 다른 의미일 경우 멀게 학습하여 텍스트-이미지를 동시에 이해할 수 있도록 적용했으며, 생성형 multi modal을 만들기 위해서는 텍스트-이미지를 하나의 벡터 시퀀스로 변환하고, 변환한 데이터를 원복 하는 기술을 적용하여, 이를 통해 텍스트-이미지에 대한 multi modal 로 확장할 수 있었다. Text와 image를 기반으로 Transformer 아키텍처 기반의 multi modal 확장 방법을 정리해 보았으나, 이러한 원리를 사용하면 이미지 뿐만 아니라 음성 데이터, 동영상 등 데이터 또한 확장 가능성이 있을 것으로 생각된다.

1. Attention Is All You Need, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, 10page, 7.Conclusion [↑](#endnote-ref-1)
2. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, 4page, 3.1 Pretraining BERT Task #1: Masked LM [↑](#endnote-ref-2)
3. AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE, Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas nterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby, 2021, 3page 3.1 VISION TRANSFORMER (ViT) [↑](#endnote-ref-3)
4. Learning Transferable Visual Models From Natural Language Supervision, Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever, 4page, 2.3 Selecting an Efficient Pre-Training Method [↑](#endnote-ref-4)
5. Zero-Shot Text-to-Image Generation, Aditya Ramesh, Mikhail Pavlov, Gabriel Goh 1, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, Ilya Sutskever, 2page, method, 13page, A.3 The Logit-Laplace Distribution [↑](#endnote-ref-5)