Ko-ViLT: Vision-Language Transformer에 기반한 한국어 시각 질의응답

1. 서론

- 최근 자연어 처리뿐만 아니라 합성곱 신경망을 활용하던 이미지 처리에도 **Transformer 구조**를 통해서 이미지 분류부터 객체 인식 작업까지 폭넓게 사용되고 좋은 성능을 보여줌

▼ Transformer

Self-Attention 계층으로 구성된 Transformer 구조는 자연어 처리 작업에 대해서 다량의 여러 문장에 대해서 가리거나 교체한 단어를 맞추도록 사전학습을 하여 여러 하위 작업에 대해서 미세 조정을 통해 좋은 성능을 달성

- 본 논문은 이미지 임베딩을 선형층을 통해 단순화하여 병목 현상을 해결하고 두 임베딩 특징을 하나의 모델에 사용하여 연산량 의 큰 이점을 주는 **ViLT**(Vision-and-Language Transformer) 모델을 활용

---

2. 관련 연구

2.1 Transformer

- Attention구조는 Query와 Key의 유사도를 구하고, 구한 유사도 가중치로 간주하여 Value와 가중합을 통해서 각 Query와 Key가 얼마나 유사한지 확인하는 구조
- Attention의 Qurey, Key, Value를 이전 계층의 은닉 상태로 한 Self-Attention 계층을 병렬적으로 연결한 MultiHeadAttention 계층에 정규화층, 앞 먹임 신경망을 결합한 블록을 디코더, 인코더에 대해 각각 여러층 쌓은 구조를 Transformer구조라 함
- 이미지 처리에서의 Transformer의 도입(ViT)이 성공을 거두자 이미지 처리의 연구 흐름은 합성곱 신경망 방법에서 Transformer 방법으로 이동하고 있음

2.2 Vision-Language Model

- 기존에는 두 영역을 한 번에 처리하기 위해 다양한 사전학습을 위한 기반 모델들이 제안됨
- 최근은 자연어 처리와, 이미지 처리 두 영역을 한 번에 처리하는 멀티모달 연구가 활발히 진행되고 있음

▼ Single-Stream 단점

이미지 임베딩이 깊어 병목현상이 발생해 연산이 느림

▼ Two-Stream 단점

모델이 무거움

> 💡 위의 단점을 해결하기 위해 Vit에서 아이디어를 얻어 Single-Stream에서 이미지 임베딩을 선형층을 통해 구하여 병목현상을 해결하는 ViLT를 제안

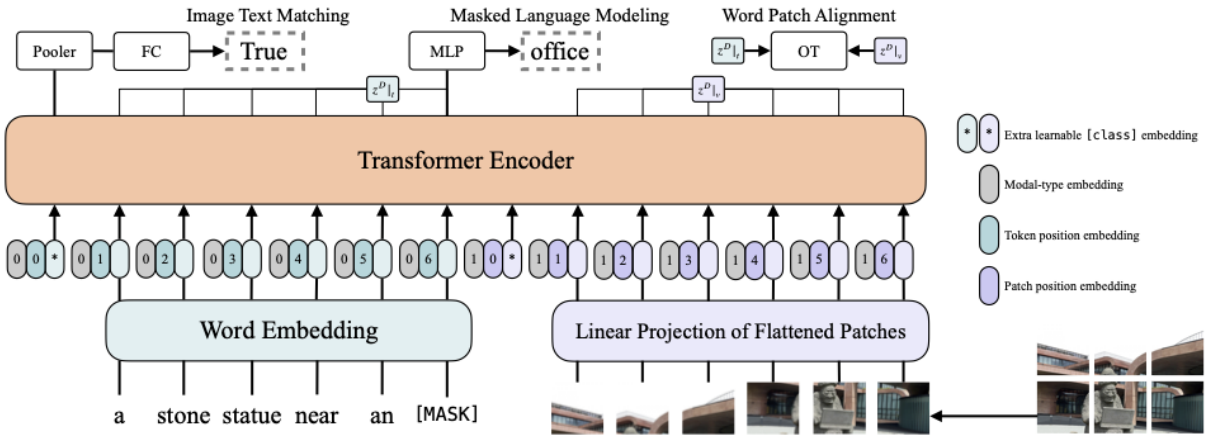

그림 1. Vision-Language Model.

3. ViLT 모델

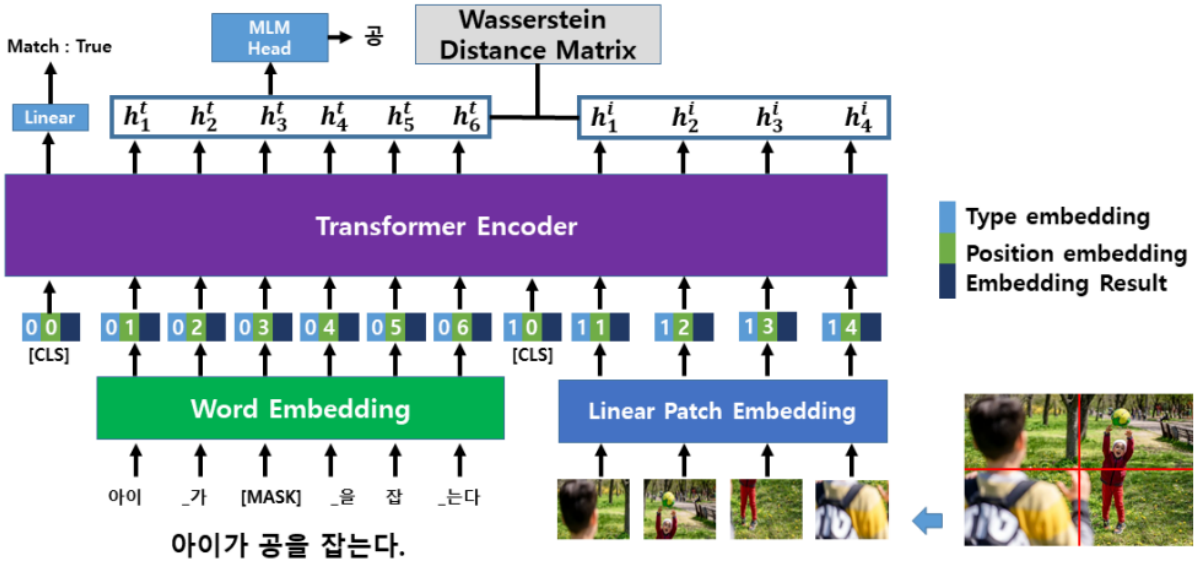

그림 1: ViLT Model 개요

그림 2. ViLT Model

- 다음 수식을 통해 은닉상태를 계산

$$\text{Emb}_{img} = \textbf{PatchEmbedding}(Patch) + V_{pos} + V_{type} \quad (1)$$

$$\text{Emb}_{text} = \textbf{WordEmbedding}(Text) + V_{pos} + V_{type} \quad (2)$$

$$h^0 = [\text{Emb}_{text}; \text{Emb}_{img}] \quad (3)$$

$$\hat{h}^d = \textbf{MHA}(\text{Linear}(h^{d-1})) + h^{d-1} \quad (4)$$

$$h^d = \textbf{MLP}(\text{Linear}(\hat{h}^d) + \hat{h}^d \quad (5)$$

단 Text는 WordPiece Tokenizer에 의해 분절된 상태

▼ 이미지와 이미지 설명을 이용해서 사전학습이 가능한데, 사전학습시 3가지 Loss를 통해서 사전 학습 함

1. 이미지 매칭
2. Masking 단어 예측
3. Wasserstein 거리를 통한 Word-Patch Alignment

Mask 예측 학습

1. 토큰의 일부를 마스킹하거나 랜덤한 단어로 변경
2. 이후 MLMHead를 통해서 단어를 예측

3. 이를 통해 단어 예측의 Loss값 L_1을 얻음
4. 이때 이미지 매칭 Loss와 Word-Patch Alignment의 Loss은 계산하지 않음

이미지 매칭

1. 배치의 절반 이미지, 설명 쌍을 섞은 후 맨 앞의 CLS 토큰을 통해서 2진 분류(일치, 불일치)
2. 이후 이 분류의 손실 값 L_2을 얻음
3. 이미지 매칭 손실 L_2를 계산하면서 Word-Patch Alignment를 학습
   a. 배치 사이즈를 B, 패치 길이 P, 문장 길이 L, 은닉 상태 차원수 H라 하면
   b. 이미지 은닉상태 행렬 $R^{(B*P*H)}$과, 문장의 은닉 상태 행렬 $R^{(B*L*H)}$의 곱을 통해서 Wasserstein 거리 행렬 $R^{(B*P*L)}$를 얻게 된다.
   c. (이미지 은닉상태 행렬 $\times$ 문장의 은닉 상태 행렬)로 text와 image간의 joint learning을 잘 수행하도록 함
      -거리를 구하는데 시간 복잡도가 크기 때문에 IPOT알고리즘을 사용하여 구함

거리가 구해지면 각 배치별로 대각합을 구해서 거리합을 구하고,

일치하는 이미지와 설명 쌍에 대해서는 최소화되도록 하여

Patch와 Word를 정렬하게 되고 이때의 Loss를 L_3라 하자.
그러면 사전학습의 손실 값은 아래의 수식처럼 구성됨.

> 💡 $L = L_1 + L_2 + 0.5 * L_3$

---

4. 실험

표 1: VQA 데이터 크기

Type	Size
Train	1,646,080
DEV(TEST)	3,000

표 1. VQA 데이터 크기

Pre-training

- MS-COCO 캡션 데이터 (600K 개)

Fine-Tuning

- 생활 및 거주환경 기반 VQA 데이터 (1.5M 개)

⚠️ VQA에서 레이블의 출현 빈도가 5번 이하의 데이터는 데이터에서 제거

Tokenizer

- Pretrained language models for korean

성능 비교

표 2: VQA 성능 비교

Test	Acc	1 Epoch Time
METER	85.24	24H
w/o pretrained ViLT	79.46	2H
w/ pretrained ViLT	78.61	2H

표 2. Two-Stream 모델인 METER와 ViLT 모델 성능비교

- 제안한 모델의 성능이 더 낮음
- 그러나, 속도에서 10배 이상 차이

▼ 비 사전학습 모델의 성능이 더 높음

1. 사전학습에 약 600K 개의 이미지-설명 데이터가 사용
2. VQA의 데이터는 약 1.5M 개의 데이터가 사용
3. 압도적인 데이터 수에 의해서 비 사전학습 모델의 성능이 높게 나올 수 있음

사전학습 유뮤에 따른 수렴속도 차이

표 3: 사전학습 유무에 따른 수렴 속도 차이

Test	Epoch1	Epoch2	Epoch3	Epoch4	Epoch5
w/o pretrained	5.21	5.21	5.21	5.21	5.28
w/ pretrained	5.21	5.21	5.21	9.14	21.95

데이터의 6%(약 80K)의 데이터를 이용해서 수렴의 속도 차이를 확인

수렴 속도가 비 사전학습 모델에 비해 압도적이므로 사전학습이 잘 되었다고 볼 수 있음.

5. 요약

- 한국어로 구축된 시각 질의응답VQA) 작업을 Vision-Language Transformer 구조를 이용해서 해결
- Two-Stream보다 성능이 소폭 낮지만, 연산시간에 대해서 10배의 이득
- 사전학습 시 성능은 더 낮지만 수렴 속도에 대해 장점을 가짐