# Transformer를 이용한 이미지 캡션 생성

이지은 ○ 박진욱 박찬희 홍정수 박상현†

연세대학교 컴퓨터과학과

{jieun199624○, parkju536, channy_12, jungsoo, sanghyun}@yonsei.ac.kr

# Image Caption Generation using Transformer

Jieun Lee○ Jinuk Park Chanhee Park Jungsoo Hong Sanghyun Park†

Dept. of Computer Science, Yonsei University

**요 약**

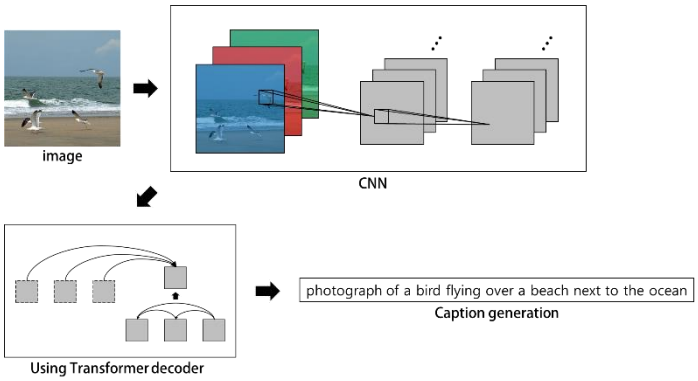
이미지 캡션 생성 기술이란 이미지를 설명하는 캡션을 자동으로 생성해내는 기술을 뜻한다. 기존의 이미지 캡션 생성 연구는 Convolutional Neural Network(CNN)을 통해 이미지 정보를 인코딩하고 Recurrent Neural Network(RNN)을 통해 캡션을 생성하는 인코더-디코더 구조를 갖는다. RNN은 순차적인 특성을 갖기 때문에 문장이 길어질수록 훈련시간이 증가한다. 본 논문은 RNN을 사용하지 않고 Multi-head 주의 집중 구조를 활용하는 Transformer를 적용함으로써 이미지의 내용을 보다 정확하게 포착하고 병렬처리가 가능하도록 하였다. MSCOCO 데이터 집합을 이용하여 비교 실험을 통해 기존 RNN을 사용하는 연구보다 개선된 성능을 입증하였다.

## 1. 서 론

이미지로부터 이미지를 설명하는 캡션을 자동으로 생성해내는 기술을 이미지 캡션 생성 기술이라고 한다. 이미지 캡션 생성 모델은 컴퓨터 비전과 관련된 이미지에서 어떤 오브젝트를 선택할 것인지에 대한 능력과, 선택한 오브젝트를 잘 포착해서 그것들 간의 관계를 이용해 자연어로 잘 표현할 수 있는 능력을 가질 수 있어야 한다.

이미지 캡션 생성은 이미지의 내용에서 캡션으로 번역하는 것이므로 번역의 한 종류라고 볼 수 있다. 이와 비슷하게 신경망 기계 번역(Neural Machine Translation)은 주어진 문장에 대해서 다른 언어로 번역을 하는 연구이다. [2]는 기계 번역 연구[3]에서 영감을 얻어 Recurrent Neural Network(RNN)를 사용해 인코더-디코더 구조(Encoder-Decoder Framework)를 채택하였다. 기계 번역과 달리 이미지의 내용을 번역하므로 인코더 부분에서는 Convolutional Neural Network(CNN)을 사용하는 차이가 있다. [1]은 인코더-디코더 구조로 CNN과 LSTM을 이용하고 [2]와 다르게 주의 집중 기법 (Attention Mechanism)을 함께 적용하였다. 주의 집중 기법을 통해 이미지의 내용이 단어를 생성하는 데에 크게 주목을 하는지 학습을 한다. [1]은 “Soft” Attention 과 “Hard” Attention을 소개하고, 두 가지의 Attention을 통해 성능을 향상시켰다.

RNN 모델은 기본적으로 이전 은닉 상태(hidden state)의 결과를 입력으로 받아 순차적으로 학습을 시킨다. 이러한 특징이 학습할 때에 병렬처리를 어렵게 하고 문장이 길어질수록 훈련하는데 걸리는 시간이 증가하게 된다. 이와 달리 Transformer[4] 모델은 RNN을 사용하는 대신 주의 집중 기법만을 이용한다. 이전 은닉 상태의 결과에 의존하지 않기 때문에 병렬처리가 가능하여 훈련시간이 크게 감소할 뿐만 아니라 향상된 성능을 보여주었다.




**[그림 1] 제안하는 모델 개요**

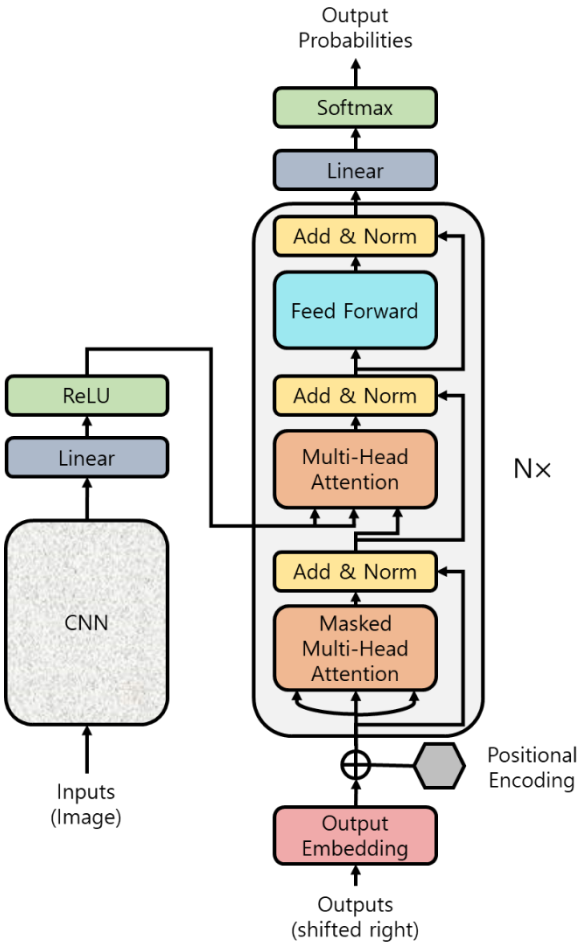
[그림 1]은 본 연구가 제안하는 모델의 개요를 나타낸다. 디코더에서 RNN을 사용하지 않고 [4]의 자가-주의 집중(Self-Attention)과 Multi-Head 주의 집중을 통해 훈련시간을 감소시키고, 단어들 간의 관계를 포함하는 정보와 이미지를 연결하여 향상된 캡션을 생성하고자 한다. 제안하는 모델의 학습 및 성능 평가를 위해서 MSCOCO 데이터 집합을 사용하여 [1]의 모델과 성능을 BLEU 점수를 통해 비교한다.

## 2. 제안 방법

본 연구에서 제안하는 구조는 [그림 2]와 같다. 인코더-디코더 구조로 이미지를 CNN을 통해 특징벡터를 추출하고 자가-주의 집중과 Multi-Head 주의 집중을 통해 캡션을 생성한다.

† 교신 저자: sanghyun@yonsei.ac.kr

* 이 논문은 과학기술정보통신부와 한국연구재단의 방사선기술개발사업으로 지원을 연구 지원한 (2017M2A2A7A02020213)의 결과물입니다.




[그림 2] 제안하는 모델 구조

### 2.1 인코더

본 연구에서는 인코더에서 CNN을 통해 이미지의 특징벡터를 추출한다. [1]과 동일하게 디코더가 이미지의 어떤 부분을 선택적으로 주목해야 하는지에 대한 정보를 전달하기위해서 CNN의 완전 연결 레이어(Fully Connected Layer)대신한 레이어 이전의 컨볼루셔널 레이어로부터 이미지의 특징벡터를 추출한다.

### 2.2 디코더

디코더에서는 RNN을 사용하는 [1]과 다르게 Multi-head 주의 집중을 사용하는 Transformer[4]의 디코더 부분을 차용하여 단어를 생성한다. RNN을 사용하지 않기 때문에 단어들의 순서를 파악할 수 없어지므로 위치 인코딩(Positional Encoding)을 통해 단어의 위치 정보를 추가한다. 디코더는 N개의 스택으로 구성되어 있고 각 레이어는 3개의 내부 레이어로 구성된다. 첫 번째 레이어로, 입력 단어들 사이의 관계에 대한 정보를 주기 위해 자가-주의 집중을 하는 Masked Multi-Head 주의 집중 레이어가 있다. Mask를 사용하여 입력 단어에서 정답이 될 다음 단어를 보지 않고 학습하기 위함이다. 두 번째 레이어로, 인코더의 결과와 첫 번째 레이어의 결과를 연결하는 Multi-Head 주의 집중 레이어가 있다. 마지막으로 ReLU 활성화 함수를 사용하고 선형 변환을 하는 전방 전달(Feed Forward) 레이어가 있다.

자가-주의 집중과 Multi-Head 주의 집중은 다음과 같이 계산된다.

$$\text{Attention}(Q,\ K,\ V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\ \ldots,\text{head}_h)W^O \quad (2)$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (3)$$

수식 (1)에서 Query(Q)와 Key(K), Value(V) 벡터를 입력으로 받아서 Query와 Key의 내적(Dot-Product)을 통해 Query가 어떤 Key와 연관성이 있는지 가중치를 구하고 Key의 차원 값으로 스케일링을 한 후에 소프트맥스(Softmax) 함수를 통해 계산한다. 그 후 Value와 곱하여 최종 주의 집중 값을 얻는다. Query와 Key, Value가 모두 같은 값이면 자가-주의 집중이라고 한다. 수식 (2)에서 Multi-Head 주의 집중은 h개의 head에서 각 주의 집중 값을 계산한후에 각 head들을 연결하여 계산한다. 수식 (3)에서 각 head는 수식 (1)과 동일하다. 모든 head들이 독립적으로 주의 집중하여 다른 내용에 대해 학습하기 때문에 보다 상세한 정보를 얻을 수 있게 된다.

## 3. 실험 및 결과

### 3.1 데이터 및 파라미터 설정

본 연구에서 제안하는 모델의 성능을 평가하기 위해 MSCOCO[5]를 사용하였다. MSCOCO는 오브젝트 검출(Object Detection)과 세그먼테이션(Segmentation), 이미지 캡션 생성 연구를 위한 대규모 데이터 집합이다. 82,783개의 훈련 이미지와 40,504개의 검증 이미지를 제공한다. 하나의 이미지마다 5문장이상 혹은 이하로 총 414,113문장으로 구성된다. 실험을 위해 5문장으로 이루어지지 않은 이미지를 제외하였다. 82,586개의 이미지 중에서 66,069개의 이미지는 학습에 사용하였고, 16,517개의 이미지는 검증에 사용하였다. 테스트 이미지로 검증 이미지에서 5문장만을 가지는 3,433개의 이미지를 사용하였다.

학습에서 사용된 파라미터로는 워드 임베딩에 512 차원을 할당하였다. 디코더는 6개의 스택을 쌓았고 Multi-Head 주의 집중은 4개의 head로 구성하였다. CNN은 ImageNet으로 pretrained InceptionV3를 사용하였고 이미지의 크기를 299x299로 조절하였다.

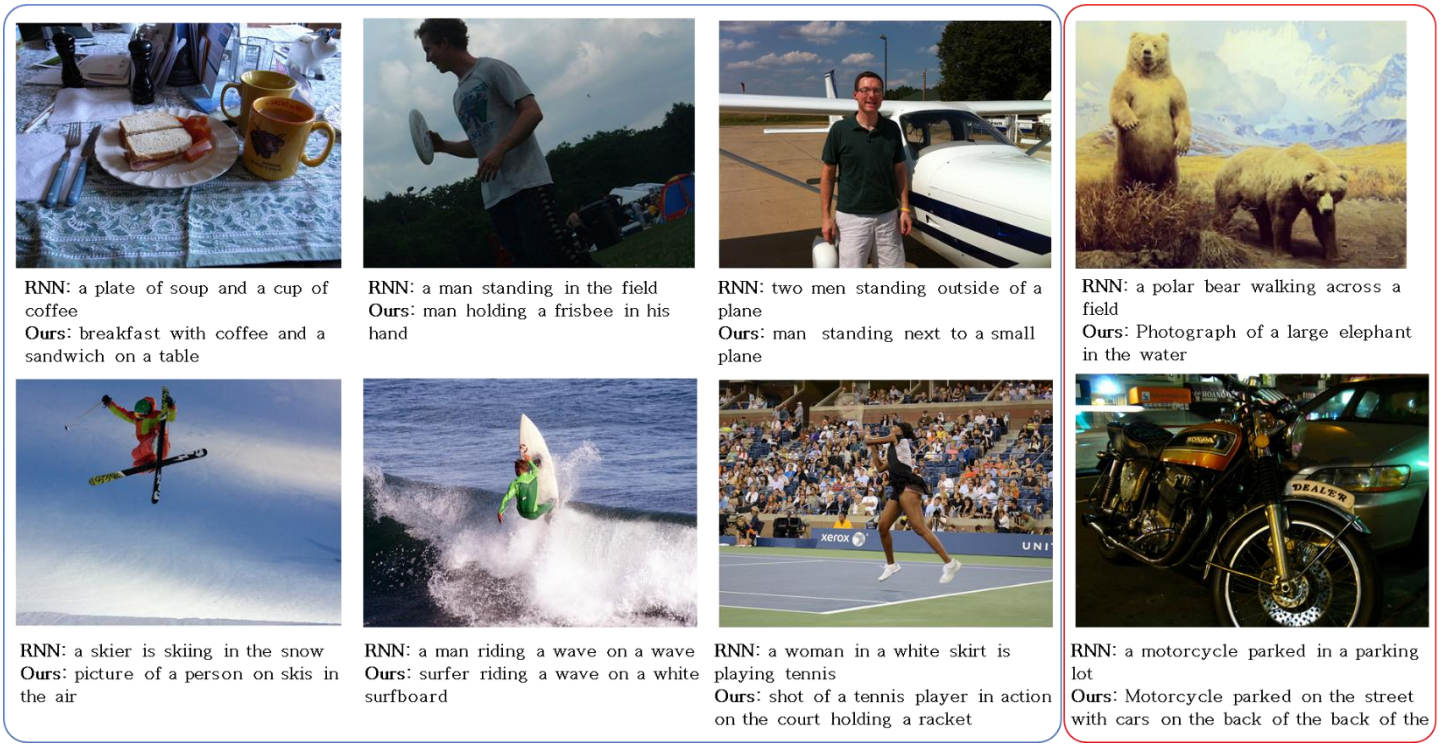
| Dataset | Model | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 |
|---|---|---|---|---|---|
| MSCOCO | RNN[1] | 0.59 | **0.42** | **0.29** | **0.20** |
| | Ours | **0.61** | 0.41 | 0.26 | 0.17 |

[표 1] 비교 실험 결과

### 3.2 실험 결과

[표 1]은 제안하는 모델과 RNN과 주의 집중 기법을 사용한 모델[1]을 비교 실험한 결과이다. 성능을 평가하는 척도로 BLEU-1, BLEU-2, BLEU-3, BLEU-4를 통해 정답 캡션과 얼마나 유사하게 생성되었는지 측정하였다.

실험 결과 전반적으로 성능이 유사함을 확인하였고 BLEU-1에서는 성능이 약 4% 향상되었다. 정량적 평가를 통해서 큰 개선이 없어 보이지만 BLEU는 n-gram을 이용해




[그림 3] 비교 실험 예시

계산하기 때문에 점수가 높더라도 결과 문장의 품질이 떨어지거나 다양성이 부족하다는 한계가 있다. [그림 3]의 파란 박스 안의 예시를 통해 정성적으로 확인한 결과, 본 연구의 모델이 Multi-head 주의 집중 기법을 통해 이미지 속 오브젝트에 대한 특징과 상황을 상세하게 추출하여 캡션을 생성하는 것을 보였다. 예를 들어, 좌측상단의 첫 번째 이미지의 경우, RNN은 'sandwich' 와 'table' 사물을 생성하지 못한 반면 제안하는 모델은 두 사물을 포함하여 상세하게 캡션을 생성하였다. 빨간색 박스는 오류가 포함된 예시로, 첫 번째 행의 그림은 이미지의 오브젝트를 잘못 포착하여 상관없는 단어를 생성했고, 아래의 행은 오브젝트를 잘 포착하였지만 중복된 단어들이 생성되었다.

훈련시간은 1 epoch마다 얼마나 시간이 소요되는지 측정하였다. RNN과 주의 집중 기법을 사용한 모델[1]은 약 6169초, 제안하는 모델은 약 3475초가 소요되었다. 결과적으로 1 epoch당 훈련시간이 약 44% 감소하였다.

## 4. 결론 및 향후 연구

본 연구에서는 이미지 캡션 생성에 RNN을 사용하지 않고 Multi-head 주의 집중 기법을 적용하여 모델을 병렬적 처리를 하고 성능이 향상된 모델을 제안하였다. MSCOCO 데이터 집합으로 학습하였고 BLEU 점수로 평가를 하였다. 기존 모델과 유사한 점수를 보였지만 정성적 실험 결과를 통해 캡션 생성 능력이 향상된 것을 검증하였다.

향후 연구에서는 중복된 단어들의 출현을 제한할 수 있는 방안과 이미지의 오브젝트를 정확하게 포착할 수 있는 방안에 대해 연구를 진행해 볼 예정이다.

## 참고문헌

[1] XU, Kelvin, et al. Show, attend and tell: Neural image caption generation with visual attention. In: *International conference on machine learning*. p. 2048-2057. 2015.

[2] VINYALS, Oriol, et al. Show and tell: A neural image caption generator. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. p. 3156-3164. 2015.

[3] SUTSKEVER, Ilya; VINYALS, Oriol; LE, Quoc V. Sequence to sequence learning with neural networks. In: *Advances in neural information processing systems*. p. 3104-3112. 2014.

[4] VASWANI, Ashish, et al. Attention is all you need. In: *Advances in neural information processing systems*. p. 5998-6008. 2017.

[5] LIN, Tsung-Yi, et al. Microsoft coco: Common objects in context. In: *European conference on computer vision*. Springer, Cham, p. 740-755. 2014.