



# Attention on Attention for Image Captioning

Lun Huang<sup>1</sup> Wenmin Wang<sup>1,3\*</sup> Jie Chen<sup>1,2</sup> Xiao-Yong Wei<sup>2</sup>

<sup>1</sup>School of Electronic and Computer Engineering, Peking University

<sup>2</sup>Peng Cheng Laboratory

<sup>3</sup>Macau University of Science and Technology

2019/ICCV

2024.02.27

이상민

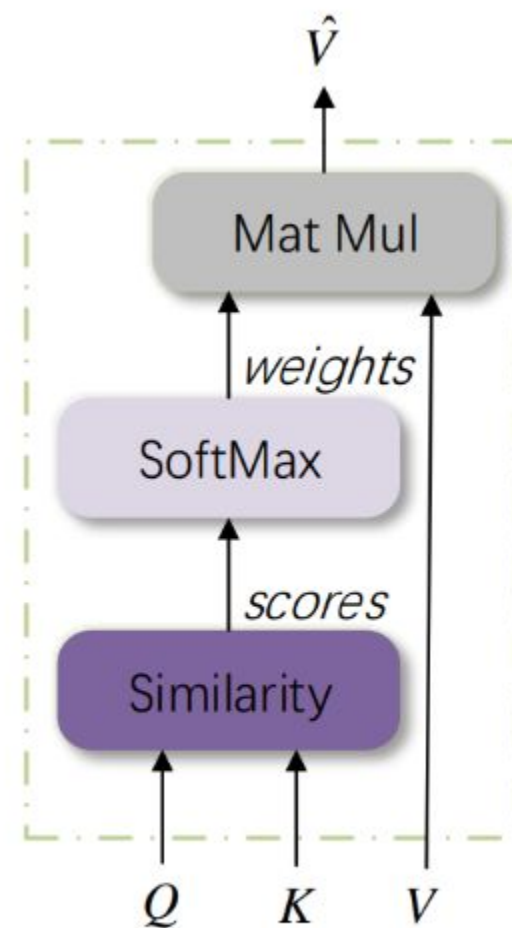
# 목차

1. Introduction
2. Method
3. Experiments
4. Conclusion

# 1. Introduction

- **image captioning에서 Attention module의 문제점**

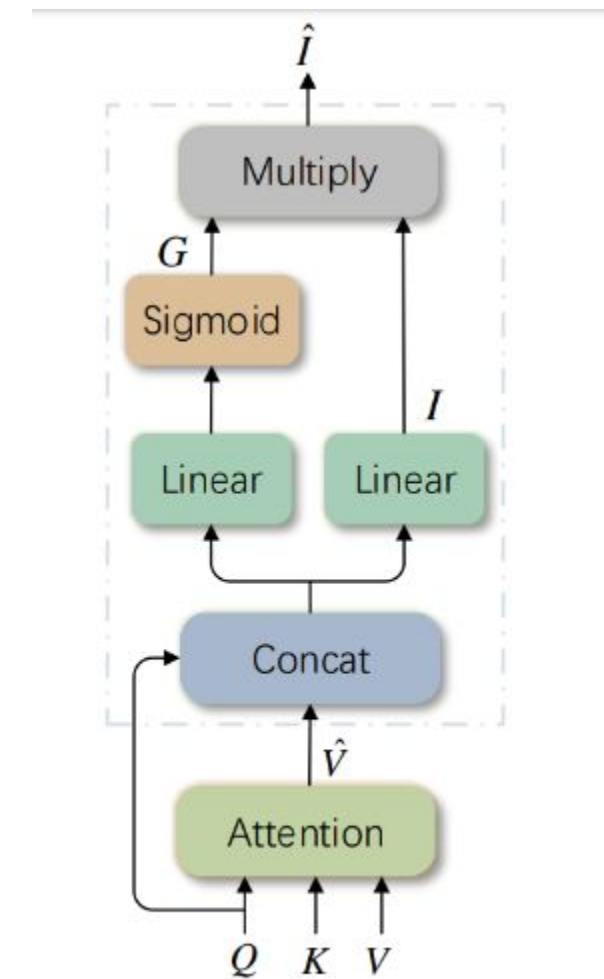
- Decoder 에서 Attention module은 이미지 정보(K/V)에서 현재 시점의 context (Query)와 연관이 있는 부분을 집중해서 참고한다
- Q와 V가 서로 관련이 없다면 decoder의 경우 현재 context연관이 없는 feature를 참고하게 되어 모델이 caption을 잘못 예측할 수 있다.



(a) Attention

# 1. Introduction

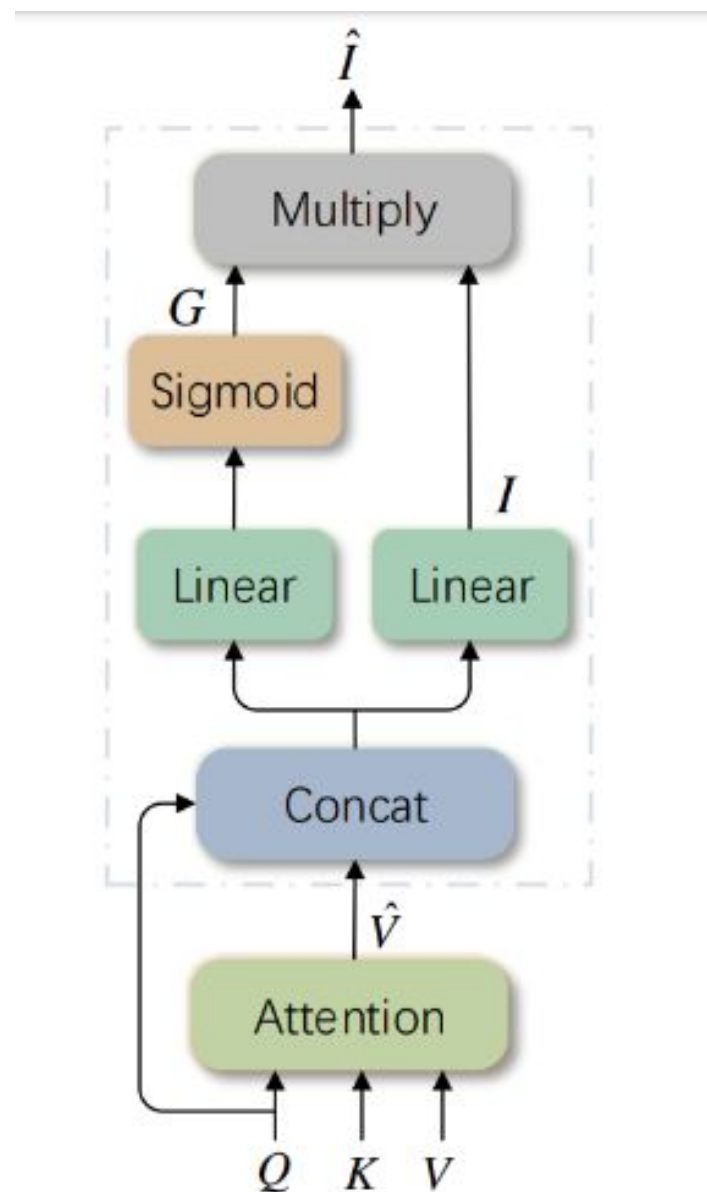
- AoA(Attention on Attention)
  - Attention module의 문제점을 개선하기 위해 제안된 module
  - Attention의 결과와 query의 관계를 파악할 수 있도록 기존 Attention module을 확장



(b) Attention on Attention

## 2. Method

- AoA mechanism



(b) Attention on Attention

### 1. Information vector & Attention gate 생성

- Attention 결과와 query를 이용해 information vector  $i$ 와 attention gate  $g$  생성

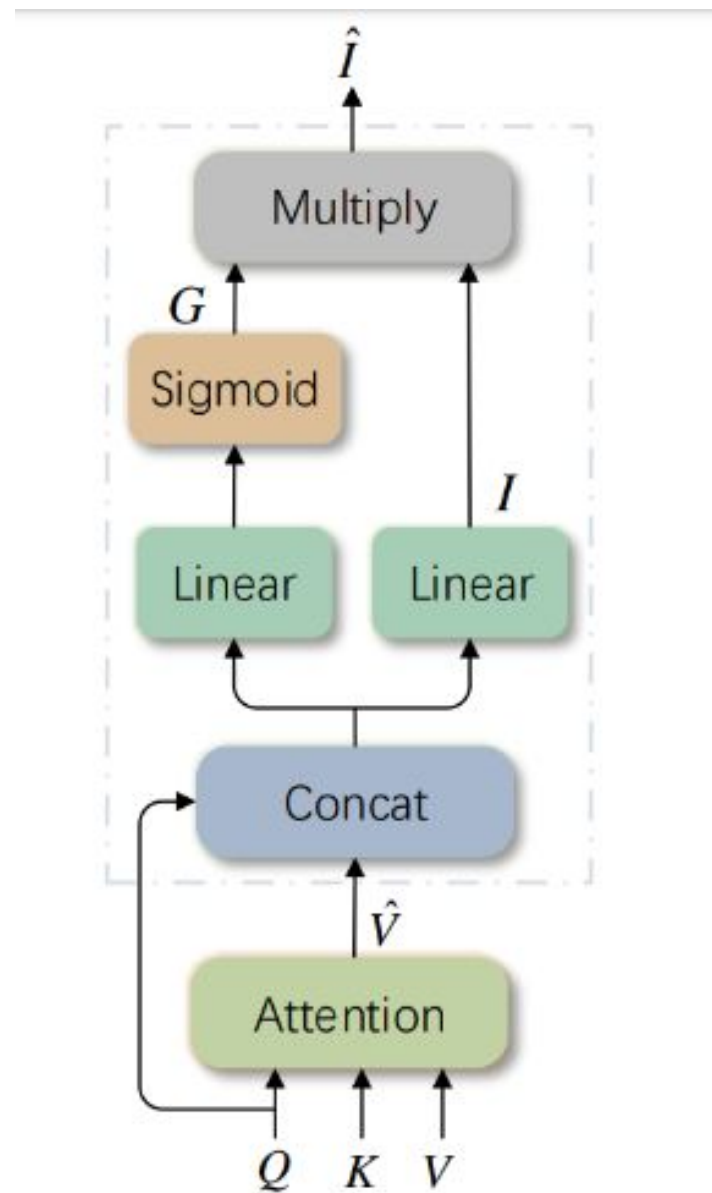
$$i = W_q^i q + W_v^i \hat{v} + b^i$$

$$g = \sigma(W_q^g q + W_v^g \hat{v} + b^g)$$

- 두 벡터는 **Linear transform** 을 통해 현재 **query (context)**와 **attention 결과**로 부터 도출.
- **Information vector**는 **attention** 으로 획득한 정보와 현재 **context(query)**의 정보를 담고 있음
- **Attention gate**의 각 **channel** 의 값은 **information vector** 에서의 해당 **channel** 의 중요성을 0~1의 수치로 나타냄

## 2. Method

- AoA mechanism



(b) Attention on Attention

### 2. 추가 Attention

- Attention gate와 information vector간 원소별 곱을 계산해 AoA의 최종 출력 결과인  $\hat{i}$  생성

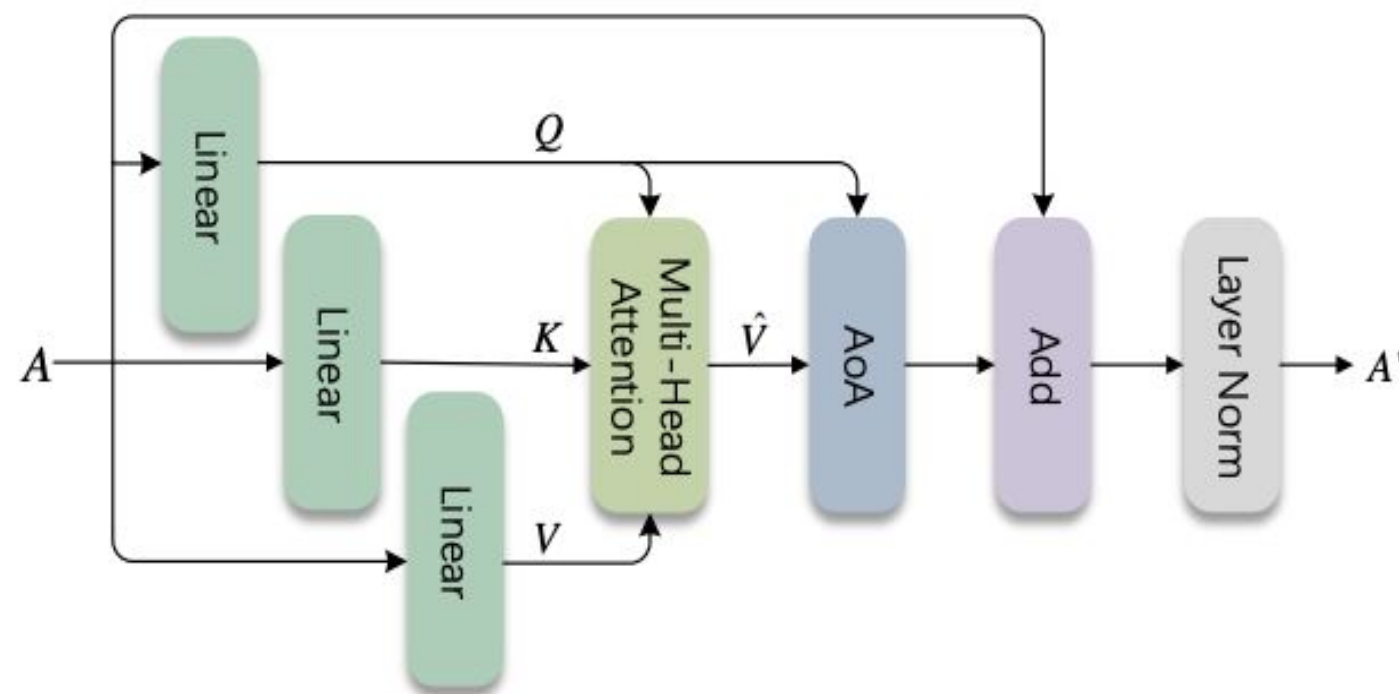
$$\hat{i} = g \odot i$$

- 원소곱을 통해 information vector에서 현재 time step에 불필요한 정보가 필터링 된다

## 2. Method

- AoANet for Image captioning

- Encoder with AoA

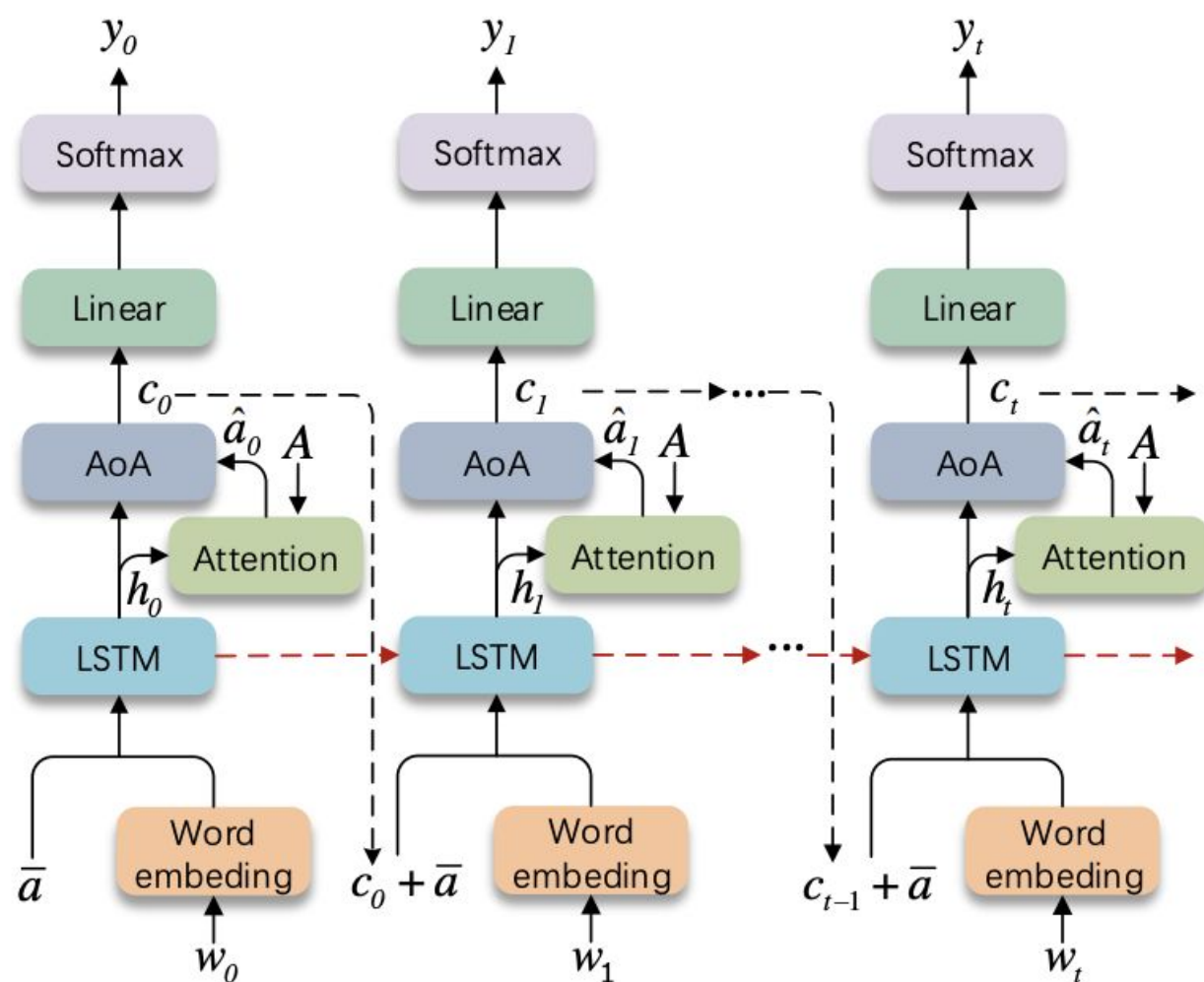


1. CNN 기반 model을 활용해 Image feature vector  $A = \{a_1, a_2, a_3, \dots, a_k\}$ 를 추출
2. 세개의 선형변환을 통해  $A$ 로 부터  $Q, K, V$ 를 생성한다.
3. Multi-head Attention layer에서  $Q, K, V$ 로 부터  $\hat{V}$ 를 생성
4. AoA layer에서는  $\hat{V}$ 와  $Q$ 로 부터 attention의 결과와 context의 정보를 담고 있는 vector를 생성
5. 이후 AoA의 출력결과는 residual connection과 Layer Norm을 거치면서 최종 Encoder의 출력이 생성된다.

## 2. Method

- AoANet for Image captioning

- Decoder with AoA



1. word embedding  $w_t$ , 이전 time step의 결과  $c_{t-1}$ 와 encoder에서 전달 받은 정보가 포함된  $\bar{a}$ 를 LSTM의 입력으로 사용
2. Attention layer에서 LSTM의 hidden state를  $Q$ , Encoder에서 받은 Image feature  $A$ 를  $K/V$ 로 attention 진행
3. AoA layer에서 Attention의 결과와 hidden state를 이용해 context vector  $c_t$  생성
4.  $c_t$ 를 linear transform한 뒤 softmax 함수에 입력해 다음 토큰의 확률 예측
5. 이과정을 순차적으로 진행해 모든 caption을 생성할 때 까지 반복



### 3. Experiments

- Quantitative Analysis

COCO test dataset에 대한 평가 결과


- BLEU-1을 제외한 모든 metric에서 가장 높은 성능을 보인다.


Model	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr-D	
Metric	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCST [31]	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.0
LSTM-A [50]	78.7	93.7	62.7	86.7	47.6	76.5	35.6	65.2	27.0	35.4	56.4	70.5	116.0	118.0
Up-Down [2]	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
RFNet [20]	80.4	95.0	64.9	89.3	50.1	80.1	38.0	69.2	28.2	37.2	58.2	73.1	122.9	125.1
GCN-LSTM [49]	-	-	65.5	89.3	50.8	80.3	38.7	69.7	28.5	37.6	58.5	73.4	125.3	126.5
SGAE [44]	<b>81.0</b>	<b>95.3</b>	65.6	89.5	50.7	80.4	38.5	69.7	28.2	37.2	58.6	73.6	123.8	126.5
AoANet (Ours)	<b>81.0</b>	95.0	<b>65.8</b>	<b>89.6</b>	<b>51.4</b>	<b>81.3</b>	<b>39.4</b>	<b>71.2</b>	<b>29.1</b>	<b>38.5</b>	<b>58.9</b>	<b>74.5</b>	<b>126.9</b>	<b>129.6</b>


### 3. Experiments

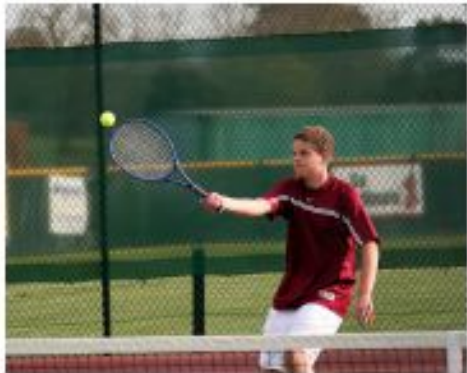
- Qualitative Analysis

- 이미지와 **baseline**, **AoANet**이 생성한 **caption** 예시

	<p><b>AoANet:</b> Two birds sitting on top of a giraffe. <b>Baseline:</b> A bird sitting on top of a tree. <b>GT1.</b> Two birds going up the back of a giraffe. <b>GT2.</b> A large giraffe that is walking by some trees. <b>GT3.</b> Two birds are sitting on a wall near the bushes.</p>
---	--

	<p><b>AoANet:</b> Two cats laying on top of a bed. <b>Baseline:</b> A black and white cat laying on top of a bed. <b>GT1.</b> A couple of cats laying on top of a bed. <b>GT2.</b> Two cats laying on a big bed and looking at the camera. <b>GT3.</b> A couple of cats on a mattress laying down.</p>
---	--

	<p><b>AoANet:</b> A cat looking at its reflection in a mirror. <b>Baseline:</b> A cat is looking out of a window. <b>GT1.</b> A cat looking at his reflection in the mirror. <b>GT2.</b> A cat that is looking in a mirror. <b>GT3.</b> A cat looking at itself in a mirror.</p>
---	--

	<p><b>AoANet:</b> A young boy hitting a tennis ball with a tennis racket. <b>Baseline:</b> A young man holding a tennis ball on a court. <b>GT1.</b> A guy in a maroon shirt is holding a tennis racket out to hit a tennis ball. <b>GT2.</b> A man on a tennis court that has a racquet. <b>GT3.</b> A boy hitting a tennis ball on the tennis court.</p>
---	--

- **Baseline** 은 생성한 **caption**은 문법에 맞지만 이미지 내용에 대해서 적절하지 못한 **caption**을

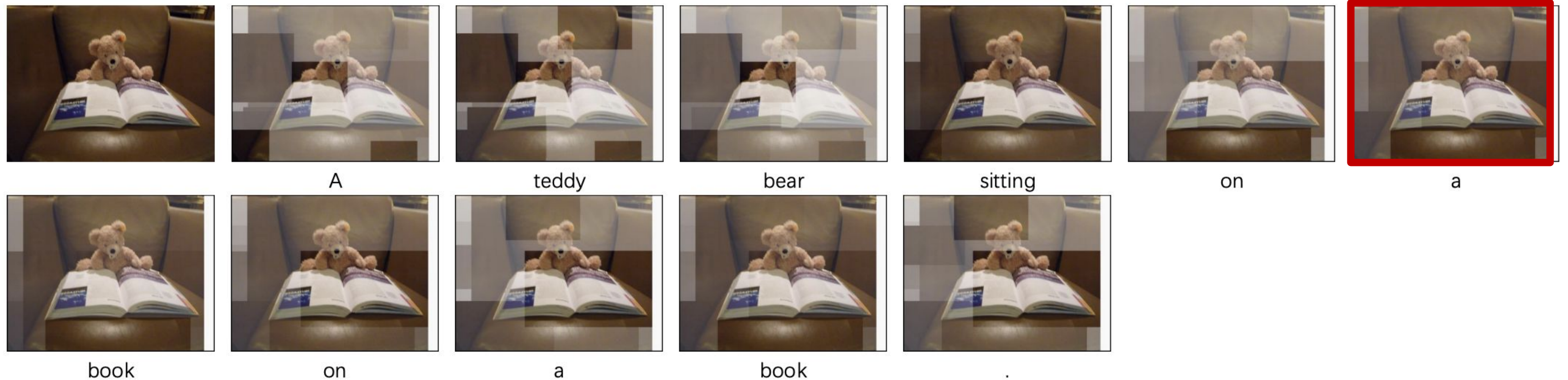
- 생성한 **AoANet**은 “boy hitting a tennis ball”, “Two birds”와 같이 이미지 내 객체의 수, 객체들 간 관계를 고려한 **caption**을 생성



### 3. Experiments

- decoding time step에서 참조된 이미지 영역 시각화

#### - baseline



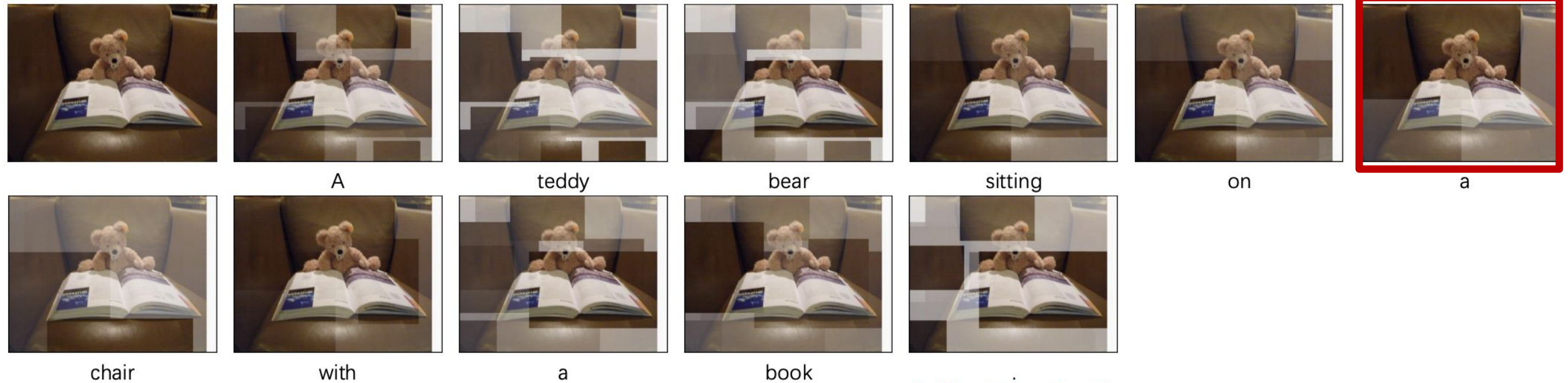
(a) Base – A teddy bear sitting **on a book on a book.**

- Baseline 모델은 “A teddy bear sitting on a”의 다음 토큰을 예측할 때 이미지에서 책이 있는 부분을 많이 참조하고 다음 토큰으로 “book”이 올 것이라 예측했다.
- 실제로 이미지에서 곰인형은 책이 아닌 의자에 앉아 있으므로 baseline은 잘못된 부분을 참조하고 있다.

### 3. Experiments

- decoding time step에서 참조된 이미지 영역 시각화

#### - AoANet



(b) AoA – A teddy bear sitting **on a chair with a book.**

- **AoANet** 모델은 “A teddy bear sitting on a”의 다음 토큰을 예측할 때 **baseline**과 달리 책이 아닌 의자를 참조하고 이미지에 맞게 다음에 올 토큰을 **chair**라고 예측했다.



## 3. Experiments

- Human Evaluation

We follow the practice in [44] and invited 30 evaluators to evaluate 100 randomly selected images. For each image, we show the evaluators two captions generated by “decoder with AoA” and the “base” model in random order, and ask them which one is more descriptive. The percentages of “decoder with AoA”, “base”, and *comparative* are 49.15%, 21.2%, and 29.65% respectively, which shows the effectiveness of AoA as confirmed by the evaluators.

AoA를 포함한 decoder가 baseline 보다 이미지를 잘 설명한다고 평가한 사람의 비율이 49.15로 가장 높다

## 4. Conclusion

- attention module을 개선하기 위해 attention result 와 context 간 연관성을 파악하는 AoA module 제안
- AoA module이 적용된 Image caption 모델 AoANet 성능 확인

# Open questions

- Attention layer를 계속 쌓으면 성능이 향상 하는 것 처럼 AoA를 계속 쌓는 것 또한 항상 성능 향상을 야기할까?