Knowing when to look: Adaptive Attention via A Visual Sentinel for Image Captioning

2019.08.29

Hyojin Ki

Paper

- Lu, Jiasen, et al. "Knowing when to look: Adaptive attention via a visual sentinel for image captioning." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

Image Captioning

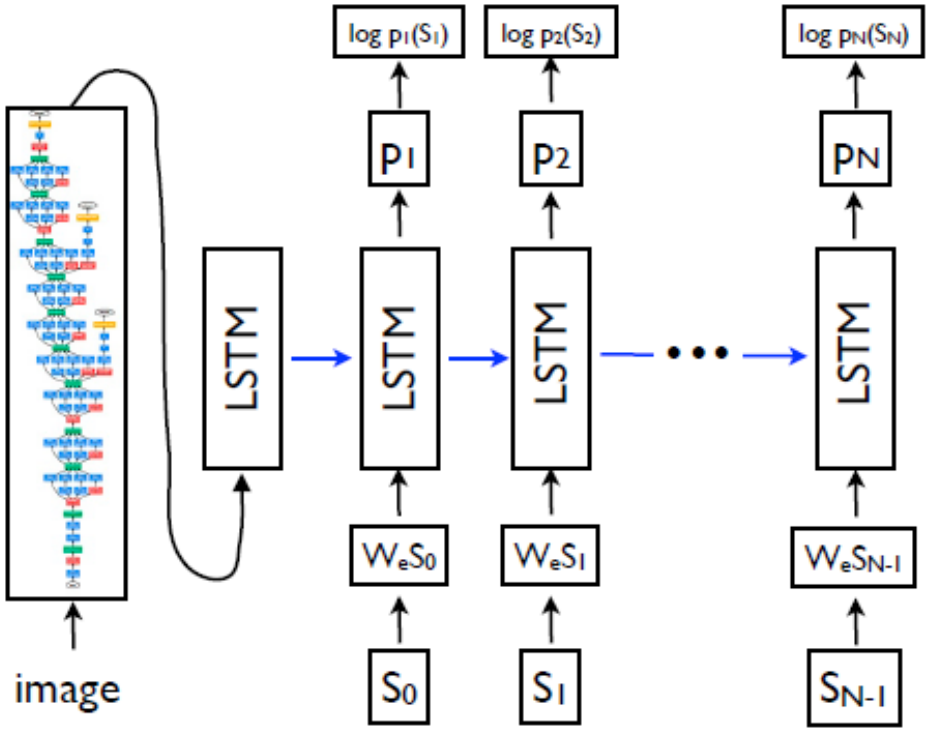


Figure 3. LSTM model combined with a CNN image embedder (as defined in [12]) and word embeddings. The unrolled connections between the LSTM memories are in blue and they correspond to the recurrent connections in Figure 2. All LSTMs share the same parameters.

Image Captioning

Attention-based neural encoder-decoder framework

- 기존의 encoder-decoder framework에 attention mechanism 사용
- 시각신호의 중요성에 상관없이 attention model이 매번 각 이미지에 적용
- 이는 자원의 낭비뿐만 아니라 비시각적 단어(a, of, on, and top, etc)를 사용한 훈련은 설명문 생성에서 성능저하 유발

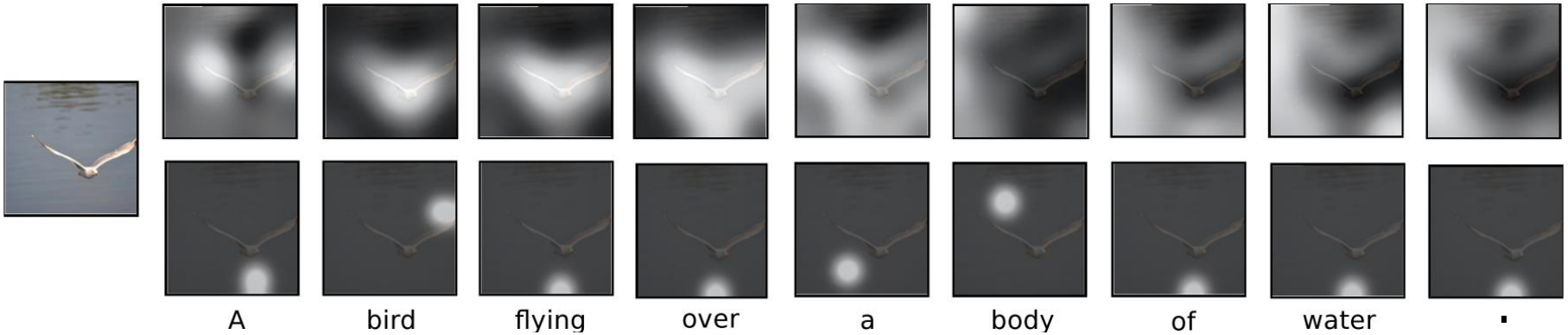
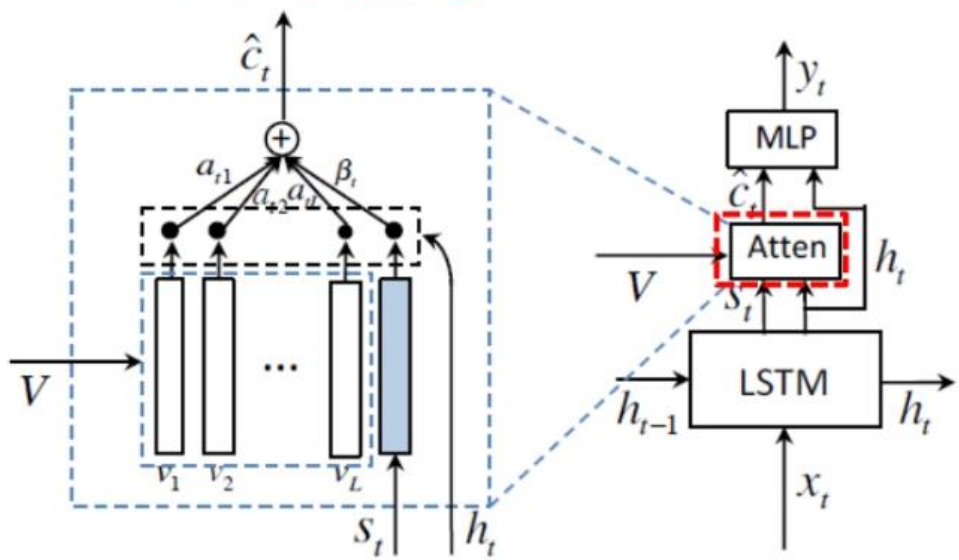


Image Captioning

Attention-based neural encoder-decoder framework

- 본 논문에서는 hidden layer위에 추가적인 시각중지 벡터(visual sentinel vector)를 갖는 새로운 adaptive attention model 제안
- 시각신호로부터 필요시 언어모델로 전환 가능
- "white", "bird", "stop",과 같은 시각적 단어에 대해서는 좀 더 이미지에 집중하고, "top", "of", "on"의 경우에는 시각중지를 사용

Atten : 새로 추가된 모듈



v_i : spatial image features S_t : visual sentinel vector

Encoder-Decoder for Image Captioning

- 사진(I)를 입력으로 주었을 때, 정답 "설명문", y를 만들어 낼 가능성을 최대가 되도록 학습 데이터 (I,y)를 이용하여 신경망의 변수 θ 들을 찾아내는 과정

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{(I,y)} \log p(y|I;\theta)$$

log likelihood of the joint probability distribution

$$\log p(y|I;\theta) = \sum_{t=0}^{N} \log p(y_t|I, y_0, y_1, \cdots, y_{t-1};\theta)$$

학습 데이터 셋(I,y)로 부터 훈련을 통해 찾아내는 변수

Encoder-Decoder for Image Captioning

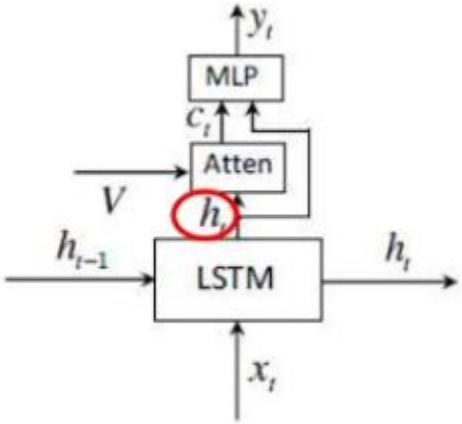
- Recurrent neural network(RNN)을 사용한 encoder-decoder framework에서의 각각의 conditional probability
- f : *nonlinear function that outputs the probability of* y_t
- h_t : *hidden state of the RNN at time t*
- c_t: *visual context vector at time t extracted from image I*

$$\log p(y_t | y_1, \dots, y_{t-1}, I) = f(h_t, c_t)$$

h_t : Long-Short Term Memory (LSTM)인 경우

$$h_t = \text{LSTM}(x_t; h_{t-1}; m_{t-1})$$

m_t : memory cell vector at time t-1



Encoder-Decoder for Image Captioning

Spatial Attention Model

1. Vanilla encoder-decoder frameworks

- c_t 는 encoder인 CNN에만 의존
- 입력 이미지, I가 CNN에 들어가서 마지막 FCL에서 global image feature 를 뽑아냄.
- 단어생성과정에서 c_t는 decoder의 hidden state에 상관없이 일정함

2. Attention based encoder-decoder frameworks

- c_t 는 encoder & decoder모두에 의존
- 시간 t에서 hidden state의 상태에 따라 decoder가 이미지의 특정 영역으로 부터의 spatial image features를 이용하여 c_t를 계산

Encoder-Decoder for Image Captioning

Spatial Attention Model

- context vector c_t는 encoder 출력(image features)과 decoder 출력(Hidden state)의 함수

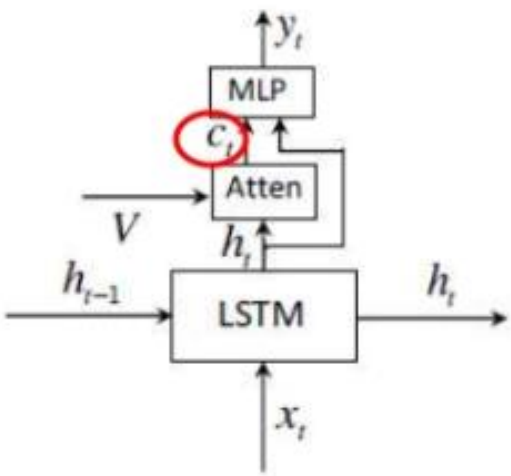
$$c_t = \boxed{g(V; h_t)}$$ attention function

$V = [v_1 \quad \cdots \quad v_k]$, $v_k \in R^d$, $V \in R^{dxk}$,

$h_t \in R^d$

v_k는 the spatial image features

h_t는 LSTM의 hidden state @ time t



Encoder-Decoder for Image Captioning

Spatial Attention Model

- 이미지상의 k개의 영역에 대한 attention 분포, α_t를 계산
- c_t, h_t는 결합되어 다음 단어 y_{t+1}를 예측

$$z_t = w_h^T tanh\left(W_v V + (W_g h_t)\mathbb{\Gamma}^T\right)$$

$$\alpha_t = \text{softmax}\ (z_t)$$

$\mathbb{\Gamma} \in R^k$ is a vector with all elements set to 1

$W_v, W_g \in R^{kxd}$ and $W_h \in R^k$

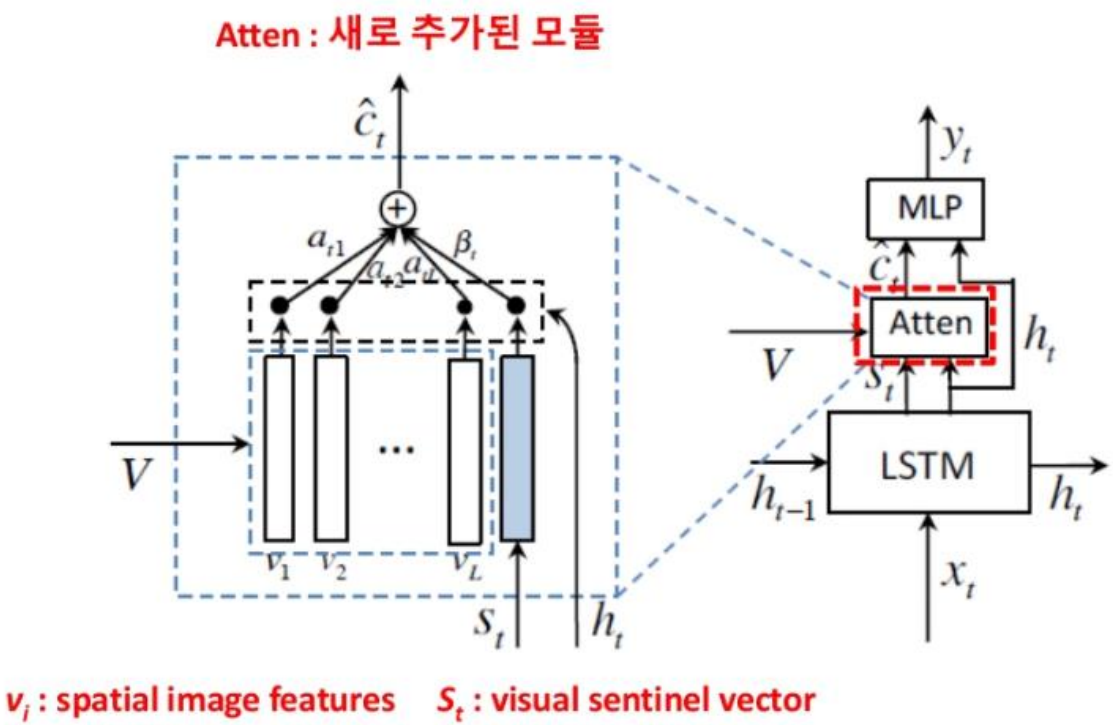
context vector c_t는 다음과 같이 정의

$$c_t = \sum_{i=1}^{k} \alpha_{ti} v_{ti}$$

Encoder-Decoder for Image Captioning

Adaptive Attention Model

• 기존의 spatial attention 기반의 decoder는 image captioning에 효과적이긴 하나, 시각적 신호와 언어모델 중 어느 쪽을 선택하여야 하는지 결정하지 못함.
-> visual sentinel gate(시각중지 게이트), β_t 를 도입



Encoder-Decoder for Image Captioning

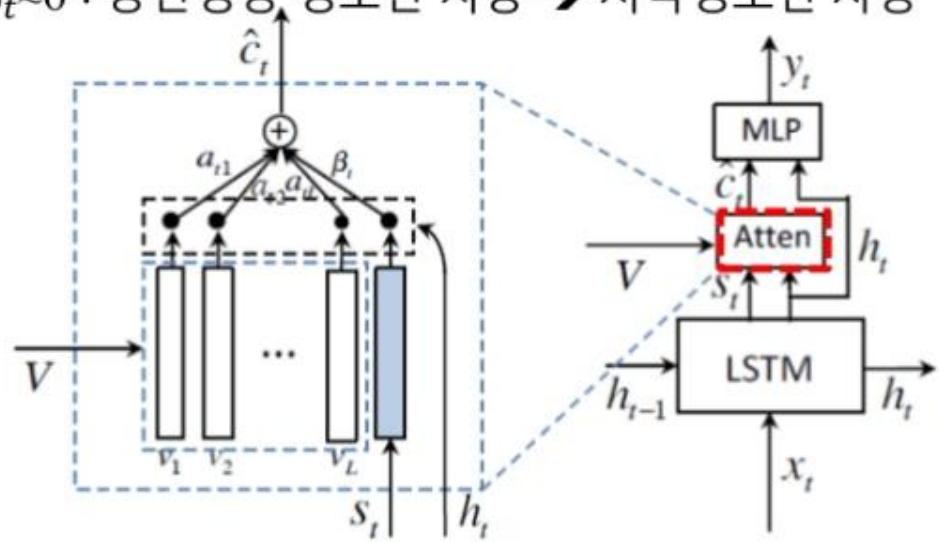
Adaptive Attention Model

- β_t 를 이용하여 새로운 Adaptive context vector $\hat{c}_t$를 정의

$$\hat{c}_t = \beta_t s_t + (1-\beta_t) c_t$$

β_t 는 0 ~ 1 사이의 상수를 생성

$\beta_t \approx 1$: 시각중지 정보만 사용 ➔ 언어모델만 사용
$\beta_t \approx 0$: 공간영상 정보만 사용 ➔ 시각정보만 사용



v_i : spatial image features S_t : visual sentinel vector

Encoder-Decoder for Image Captioning

Adaptive Attention Model

- 기존의 이미지 상의 attention 분포, α_t를 다음과 같이 수정

$$\widehat{\alpha_t}=\text{softmax}\left(\left[z_t;\ w_h^T tanh\left(W_s s_t + W_g h_t\right)\right]\right)$$

$\widehat{\alpha_t} \in R^{k+1}$로 spatial image feature($R^k$)와 visual sentinel vector 에 대한 attention 분포를 나타냄.

여기서는 $\widehat{\boldsymbol{\alpha}}_t(\boldsymbol{k}+\mathbf{1})$, 즉 $\widehat{\boldsymbol{\alpha}}_t$의 마지막 값을 $\boldsymbol{\beta}_t$값**으로 정의 함.**

시간 t에서 가능한 단어들의 어휘에 대한 선택 확률은 다음과 같이 계산

$$p_t=\text{softmax}\left(W_p(\widehat{c_t} + h_t)\right)$$

Encoder-Decoder for Image Captioning

Encoder-CNN

- 이미지에 대한 encoding은 ResNet의 마지막 convolutional layer의 spatial feature 출력(2,048X7X7)을 사용
- 이미지상의 k개의 영역의 CNN의 spatial feature과의 관계

$$A = [a_1 \quad \dots \quad a_k],\ a_i \in R^{2,048}$$

Global image feature, a^g는 다음과 같이 평균값으로 정의

$$a^g = \frac{1}{k}\sum_{i=1}^{k} a_i$$

Encoder-Decoder for Image Captioning

Encoder-CNN

a_i 로 부터 d차원의 v_i로 변환하기

$$v_i = \text{ReLU}\,(W_a a_i)$$

$$v_g = \text{ReLU}\,(W_b a^g)$$

Decoder-RNN

입력벡터, x_t

$$x_t = [w_t; v_g]$$

w_t는 “word embedding vector”, v_g는 "global image feature vector”

Experiment result

Method	Flickr30k						MS-COCO					
	B-1	B-2	B-3	B-4	METEOR	CIDEr	B-1	B-2	B-3	B-4	METEOR	CIDEr
DeepVS [11]	0.573	0.369	0.240	0.157	0.153	0.247	0.625	0.450	0.321	0.230	0.195	0.660
Hard-Attention [30]	0.669	0.439	0.296	0.199	0.185	-	0.718	0.504	0.357	0.250	0.230	-
ATT-FCN† [34]	0.647	0.460	0.324	0.230	0.189	-	0.709	0.537	0.402	0.304	0.243	-
ERD [32]	-	-	-	-	-	-	-	-	-	0.298	0.240	0.895
MSM† [33]	-	-	-	-	-	-	0.730	0.565	0.429	0.325	0.251	0.986
Ours-Spatial	0.644	0.462	0.327	0.231	0.202	0.493	0.734	0.566	0.418	0.304	0.257	1.029
Ours-Adaptive	**0.677**	**0.494**	**0.354**	**0.251**	**0.204**	**0.531**	**0.742**	**0.580**	**0.439**	**0.332**	**0.266**	**1.085**

Table 1: Performance on Flickr30k and COCO test splits. † indicates ensemble models. **B-n** is BLEU score that uses up to n-grams. Higher is better in all columns. For future comparisons, our ROUGE-L/SPICE Flickr30k scores are 0.467/0.145 and the COCO scores are 0.549/0.194.

Method	B-1		B-2		B-3		B-4		METEOR		ROUGE-L		CIDEr	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Google NIC [27]	0.713	0.895	0.542	0.802	0.407	0.694	0.309	0.587	0.254	0.346	0.530	0.682	0.943	0.946
MS Captivator [8]	0.715	0.907	0.543	0.819	0.407	0.710	0.308	0.601	0.248	0.339	0.526	0.680	0.931	0.937
m-RNN [18]	0.716	0.890	0.545	0.798	0.404	0.687	0.299	0.575	0.242	0.325	0.521	0.666	0.917	0.935
LRCN [7]	0.718	0.895	0.548	0.804	0.409	0.695	0.306	0.585	0.247	0.335	0.528	0.678	0.921	0.934
Hard-Attention [30]	0.705	0.881	0.528	0.779	0.383	0.658	0.277	0.537	0.241	0.322	0.516	0.654	0.865	0.893
ATT-FCN [34]	0.731	0.900	0.565	0.815	0.424	0.709	0.316	0.599	0.250	0.335	0.535	0.682	0.943	0.958
ERD [32]	0.720	0.900	0.550	0.812	0.414	0.705	0.313	0.597	0.256	0.347	0.533	0.686	0.965	0.969
MSM [33]	0.739	0.919	0.575	0.842	0.436	0.740	0.330	0.632	0.256	0.350	0.542	0.700	0.984	1.003
Ours-Adaptive	**0.748**	**0.920**	**0.584**	**0.845**	**0.444**	**0.744**	**0.336**	**0.637**	**0.264**	**0.359**	**0.550**	**0.705**	**1.042**	**1.059**

Table 2: Leaderboard of the published state-of-the-art image captioning models on the online COCO testing server. Our submission is a ensemble of 5 models trained with different initialization.

Experiment result

Figure 4: Visualization of generated captions and image attention maps on the COCO dataset. Different colors show a correspondence between attended regions and underlined words. First 2 columns are success cases, last columns are failure examples. Best viewed in color.

Experiment result

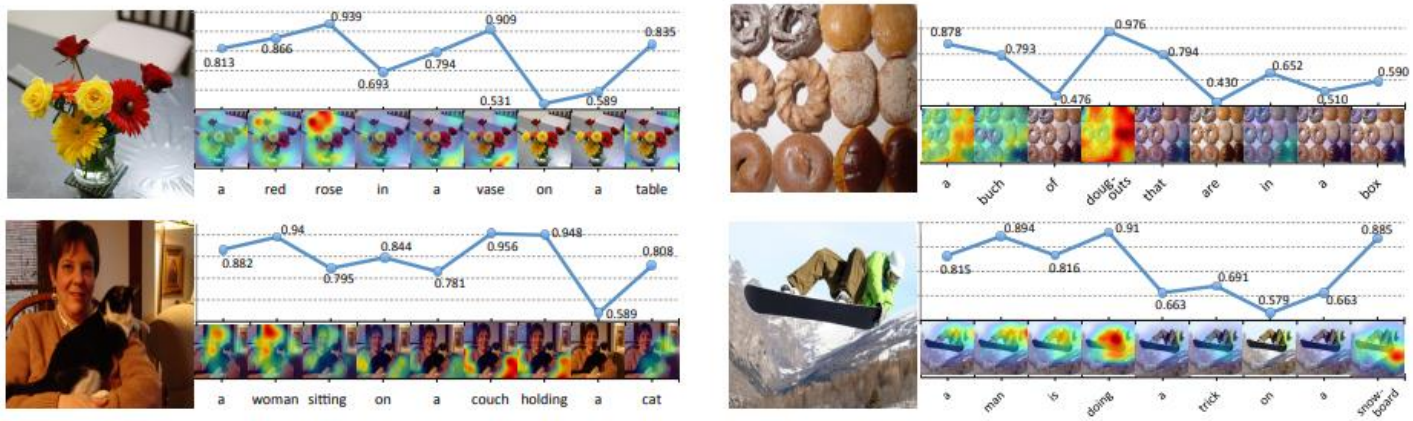


Figure 5: Visualization of generated captions, visual grounding probabilities of each generated word, and corresponding spatial attention maps produced by our model.

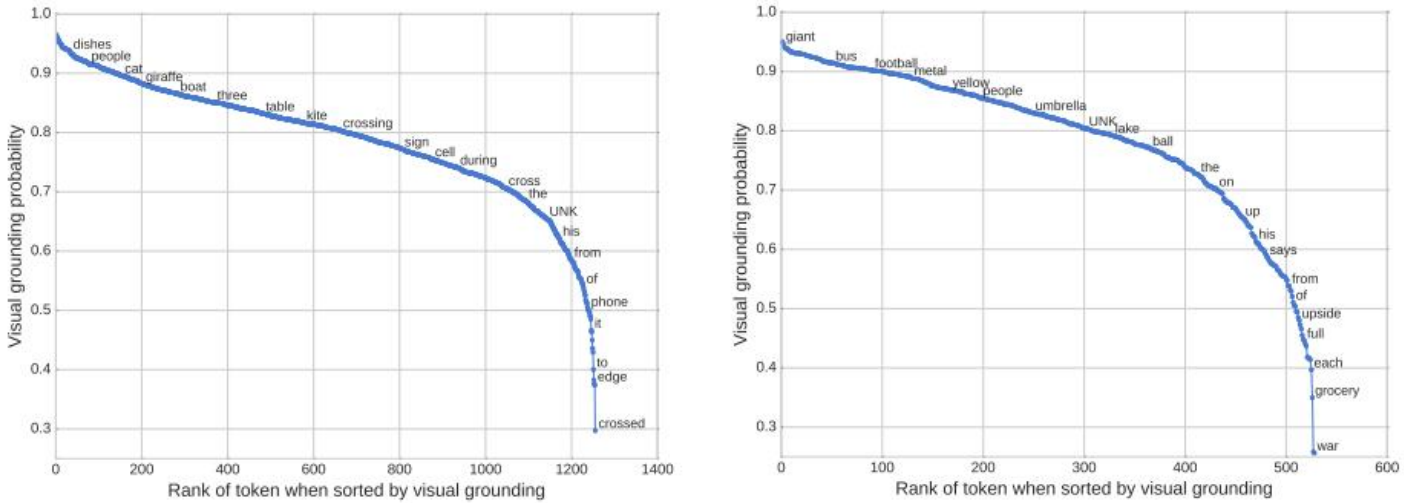


Figure 6: Rank-probability plots on COCO (left) and Flickr30k (right) indicating how likely a word is to be visually grounded when it is generated in a caption.

Experiment result

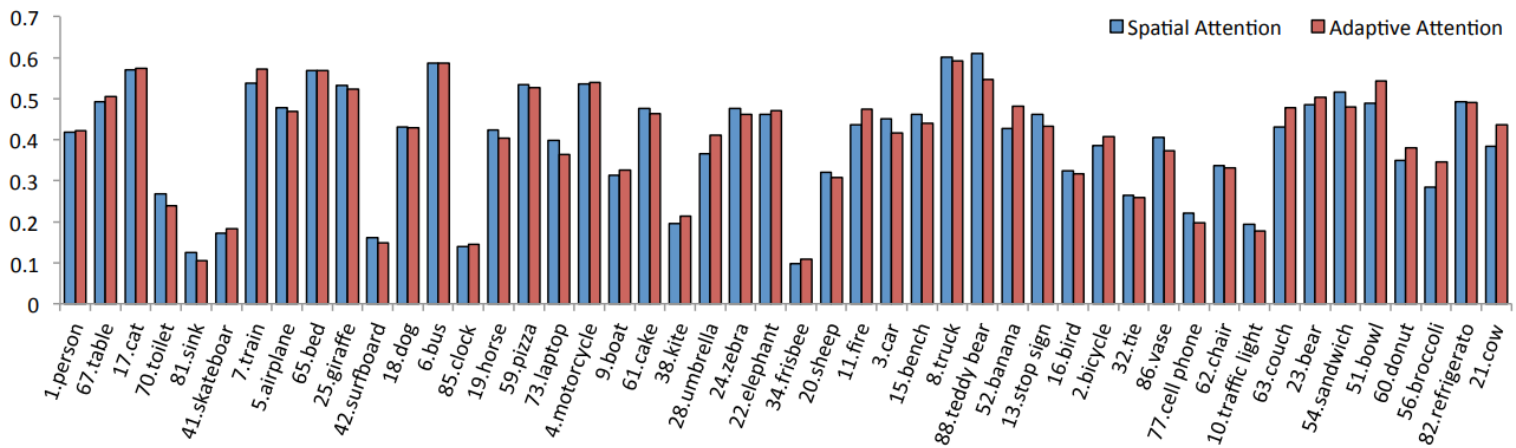


Figure 7: Localization accuracy over generated captions for top 45 most frequent COCO object categories. "Spatial Attention" and "Adaptive Attention" are our proposed spatial attention model and adaptive attention model, respectively. The COCO categories are ranked based on the align results of our adaptive attention, which cover 93.8% and 94.0% of total matched regions for spatial attention and adaptive attention, respectively.

Conclusion

- In this paper, we present a novel adaptive attention encoder-decoder framework, which provides a fallback option to the decoder. We further introduce a new LSTM extension, which produces an additional "visual sentinel".

- Our model achieves state-of-the-art performance across standard benchmarks on image captioning.

- We perform extensive attention evaluation to analysis our adaptive attention.

- Though our model is evaluated on image captioning, it can have useful applications in other domains

Paper

Reference

- https://www.slideshare.net/ssuser06e0c5/knowing-when-to-look-adaptive-attention-via-a-visual-sentinel-for-image-captioning
- https://arxiv.org/pdf/1612.01887.pdf

Thank You!

Do you have any question?