Image Captioning 논문 리뷰

Show and tell :
A Neural Image Caption Generator

KUBIG 이미지 스터디 김지후

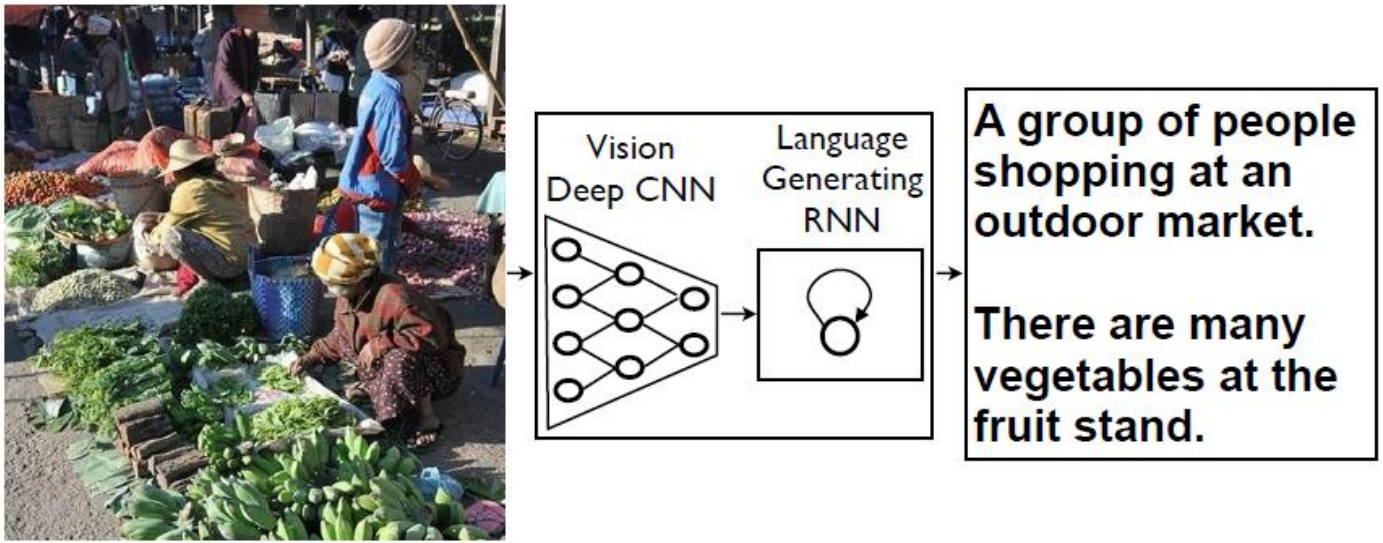
Image Captioning

이미지의 내용을 설명하는 문장을 생성하는 것으로
CV와 NLP를 연결하는 인공지능 분야

물체인식 + 특성, 활동, 관계 인식 + 자연어표현

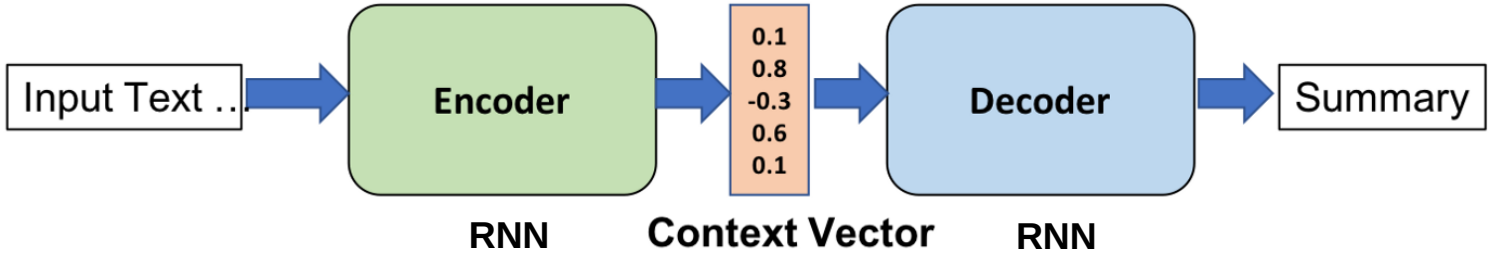

Show and tell :
A Neural Image Caption Generator

2015년 Google에서 발표한 논문
single joint model인 **NIC(Neural Image Caption Generator)** 제시

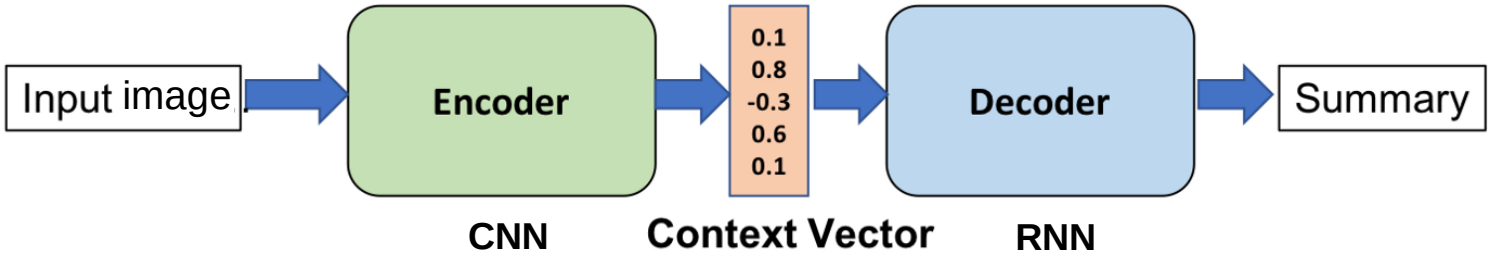


01 Introduction

기계번역의 구조



NIC 구조



02 Model

다음의 식을 사용해, **입력 이미지**가 주어졌을 때 **정답 묘사(correct description)**이 나올 확률을 directly **maximize**

$$\theta^{\star} = \arg\max_{\theta} \sum_{(I,S)} \log p(S|I;\theta)$$

θ : 모수
I : 이미지
S : 정답 묘사(correct description)

02 Model

S는 어떠한 **문장**이기 때문에 **길이가 제한되어 있지 않음**.

--> 따라서 **chain rule**을 사용해 S_0, ..., S_N까지의 확률을 **결합확률**로 나타냄

$$\log p(S|I) = \sum_{t=0}^{N} \log p(S_t|I, S_0, \ldots, S_{t-1})$$

N : 문장 길이

train 과정 : 이미지와 정답 묘사가 **(S,I)의 pair**로 주어지고,
로그 확률의 합에 따라 SGD를 사용해 최적화

02 Model

이 확률(P)을 RNN으로 구현

$$h_{t+1} = f(h_t, x_t) \ .$$

t-1개의 단어가 고정된 길이의 **hidden state(h_t)로 표현**

h_t는 새로운 input x_t이 들어오면

non-linaer function, **f를 통해 업데이트**

02 Model

1. 구체적으로 어떤 f를 사용할까?

f는 번역과 같은 sequence task에 높은 성능을 보여준 **LSTM** net을 사용

2. 이미지와 단어가 어떻게 input x_t로 들어갈까?

이미지를 표현하기 위해서 object recognition과 detection에서 좋은 성능을 보여준 **CNN**을 사용

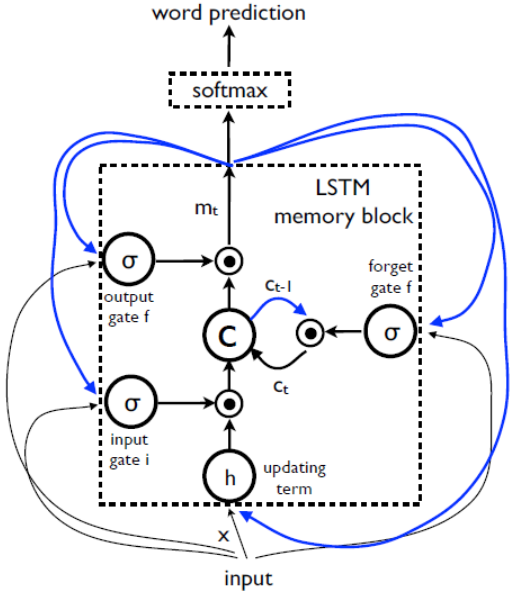
단어는 **embedding model**로 표현되었습니다.

02 Model

1) LSTM based Sentence Generator

RNN의 미분사라짐과 미분폭발을 방지하기 위해서 **LSTM**을 사용

매 input 마다 업데이트 되는 memory cell C는 3개의 gate가 각각 곱해지며 통제



$$
\begin{aligned}
i_t &= \sigma(W_{ix}x_t + W_{im}m_{t-1}) \\
f_t &= \sigma(W_{fx}x_t + W_{fm}m_{t-1}) \\
o_t &= \sigma(W_{ox}x_t + W_{om}m_{t-1}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1}) \\
m_t &= o_t \odot c_t \\
p_{t+1} &= \mathrm{Softmax}(m_t)
\end{aligned}
$$

02 Model

(1) training

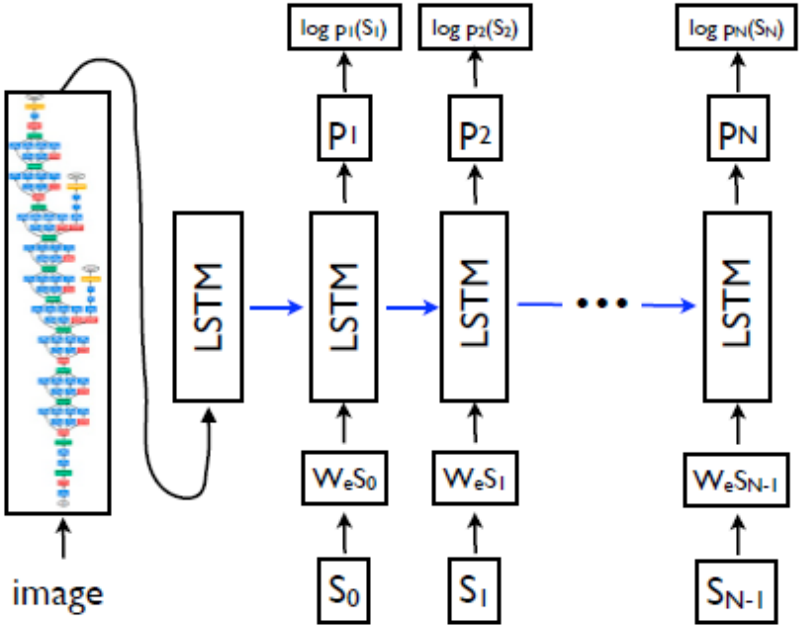
LSTM 모델은 **이미지(I)와 이전의 단어들(S_0, … , S_t-1)을 사용해 문장의 다음 단어를 예측**. t time일 때, 각각의 LSTM은 같은 모수와 output m_t-1을 공유.

$$
\begin{aligned}
x_{-1} &= \mathrm{CNN}(I) \\
x_t &= W_e S_t, \quad t \in \{0 \dots N-1\} \\
p_{t+1} &= \mathrm{LSTM}(x_t), \quad t \in \{0 \dots N-1\}
\end{aligned}
$$

I : image
W_e : word embedding
S_t : 각각 dictionary size의 차원인 one-hot vector

02 Model

(1) training



첫번째 input은 **CNN(I)**이고,

여기서 나온 **hidden state**가 두번째 **input S_0**와 합쳐져

결과적으로 **output S_1과 새로운 hidden state**를 만듦.

- 문장의 시작과 끝을 구분

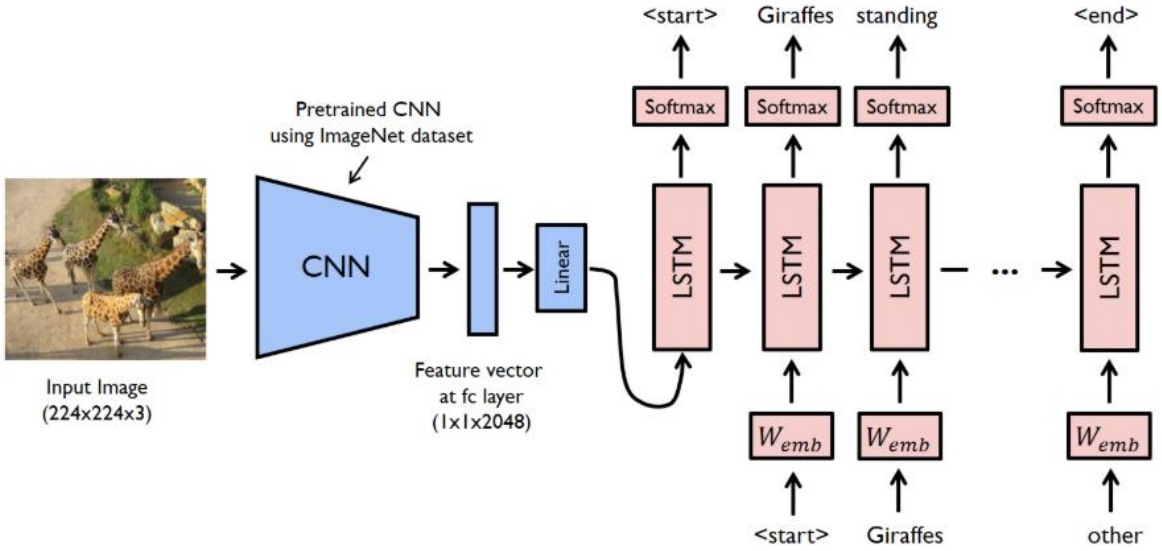
S_0: special start word

S_N: special stop word

02 Model

(1) training

각각의 단어 S에 **word embedding W_e**를 곱해 CNN을 통한 **image representation과 word가 같은 차원**에 있도록 함.



02 Model

(1) training

손실함수는 다음과 같이 각 step에서 correct word의 음의 로그 우도함수입니다.

이를 **최소화 시키는 방향으로 LSTM의 parameter와 word embedding W_e, CNN의 image embedding을 하는 top layer**를 학습합니다.

$$L(I, S) = -\sum_{t=1}^{N} \log p_t(S_t) \ .$$

02 Model

(2) Inference

문장을 만드는 다양한 방법

1. Sampling

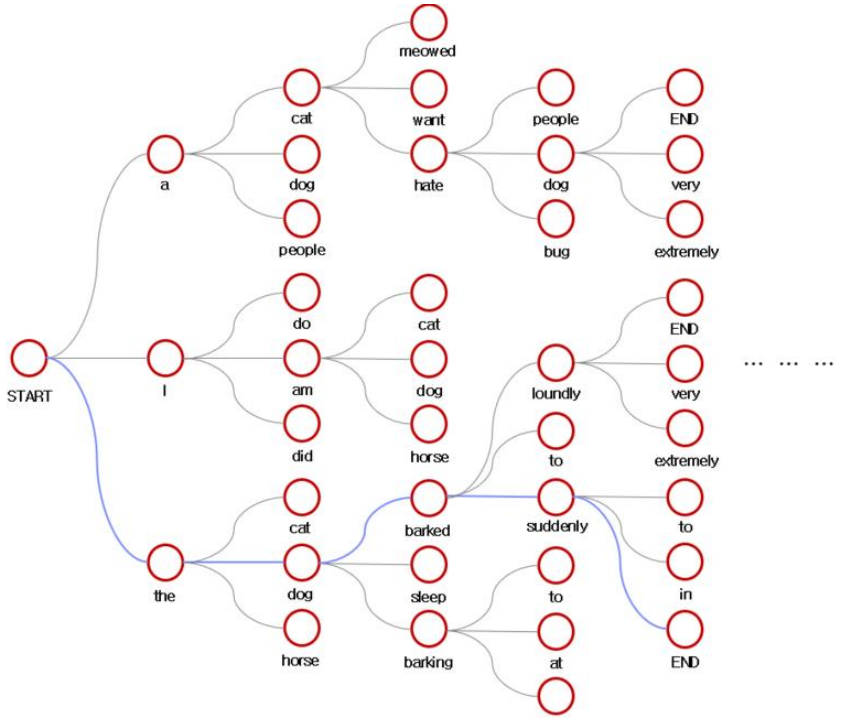
p_1에 따라 첫번째 단어를 만들고, 이를 다시 input으로 해 p_2를 만들고, 이 과정을 special end-of-sentence token이 나올 때까지 또는 최대길이까지 반복

02 Model

(2) Inference

2. BeamSearch

매 time t까지의 k개의 best sentences set을 사이즈가 t+1인 **문장을 생성할 후보군으로** 고려합니다. 이 후보로 만든 t+1개의 문장 중 **best k개의 문장을 반환합니다**. 즉 상위 k개를 선택하고 이 과정을 반복하는 것이 BeamSearch입니다.



03 Experiments

1) Generation Results

Approach	PASCAL (xfer)	Flickr 30k	Flickr 8k	SBU
Im2Text [24]				11
TreeTalk [18]				19
BabyTalk [16]	25			
Tri5Sem [11]			48	
m-RNN [21]		55	58	
MNLM [14][5]		56	51	
SOTA	25	56	58	19
NIC	**59**	**66**	**63**	**28**
Human	69	68	70	

Table 2. BLEU-1 scores. We only report previous work results when available. SOTA stands for the current state-of-the-art.

BLEU는 reference sentence와 얼마나 비슷한가를 평가하는 척도로, 구체적으로 몇개의 n-gram이 reference sentence와 겹치는 정도를 평가합니다.

BLEU Score 값 비교결과 **NIC가 가장 좋은 성능을** 보여줍니다.

04 Conclusion

A person riding a motorcycle on a dirt road.

Two dogs play in the grass.

A skateboarder does a trick on a ramp.

A dog is jumping to catch a frisbee.

A group of young people playing a game of frisbee.

Two hockey players are fighting over the puck.

A little girl in a pink hat is blowing bubbles.

A refrigerator filled with lots of food and drinks.

A herd of elephants walking across a dry grass field.

A close up of a cat laying on a couch.

A red motorcycle parked on the side of the road.

A yellow school bus parked in a parking lot.

Describes without errors	Describes with minor errors	Somewhat related to the image	Unrelated to the image

04 Conclusion

NIC는 이미지를 compact representation으로 encode하는 **Convolution neural network(CNN**)과
그에 대응하는 문장을 생성하는 **Recurrent neural network(RNN)**에 기반합니다.

실험 결과 질적으로 합리적인 문장을 생성했고 ranking metrics나 BLEU로 측정했을 때 양적으로 좋은 성능을 보여줬습니다.

◀ ▶

감사합니다