Show, Attend and Tell 모델을 이용한 한국어 캡션 생성

*김다솔 **이계민
서울과학기술대학교 전자IT미디어공학과
*dosol524@gmail.com **gyemin@seoultech.ac.kr

Korean Image Caption Generator Based on Show, Attend and Tell Model

*Kim, Dasol **Lee, Gyemin
Electronic & IT Media Engineering, Seoul National University of Science and Technology

요 약

최근 딥러닝 기술이 발전하면서 이미지를 설명하는 캡션을 생성하는 모델 또한 발전하였다. 하지만 기존 이미지 캡션 모델은 대다수 영어로 구현되어있어 영어로 캡션을 생성하게 된다. 따라서 한국어 캡션을 생성하기 위해서는 영어 이미지 캡션 결과를 한국어로 번역하는 과정이 필요하다는 문제가 있다. 이에 본 연구에서는 기존의 이미지 캡션 모델을 이용하여 한국어 캡션을 직접 생성하는 모델을 만들고자 한다. 이를 위해 이미지 캡션 모델 중 잘 알려진 Show, Attend and Tell 모델을 이용하였다. 학습에는 MS-COCO 데이터의 한국어 캡션 데이터셋을 이용하였다. 한국어 형태소 분석기를 이용하여 토큰을 만들고 캡션 모델을 재학습하여 한국어 캡션을 생성할 수 있었다. 만들어진 한국어 이미지 캡션 모델은 BLEU 스코어를 사용하여 평가하였다. 이때 BLEU 스코어를 사용하여 생성된 한국어 캡션과 영어 캡션의 성능을 평가함에 있어서 언어의 차이에 인한 결과 차이가 발생할 수 있으므로, 영어 이미지 캡션 생성 모델의 출력을 한국어로 번역하여 같은 언어로 모델을 평가한 후 최종 성능을 비교하였다. 평가 결과 한국어 이미지 캡션 생성 모델이 영어 이미지 캡션 생성 모델을 한국어로 번역한 결과보다 좋은 BLEU 스코어를 갖는 것을 확인할 수 있었다.

1. 서론

정보의 바다 속에서 필요한 데이터를 찾는 일은 여간 쉽지 않다. 텍스트 데이터는 문자로 검색하여 원하는 정보를 찾을 수 있는 반면, 이미지와 영상 데이터에서 필요한 정보를 검색하는 일은 더더욱 어려운 일이다. 그리고 데이터의 품질이 발전하면서 데이터의 양 또한 증가하고 있기에, 더 많은 데이터를 수집, 분석 하는 일은 점차 더 어려워진다.

이미지 캡셔닝은 이러한 상황에 도움을 줄 수 있는 기술이다. 이미지 캡셔닝이란 이미지를 입력하면 이미지를 설명하는 캡션을 만들어주는 기술이다. 이미지 내에 있는 객체에 대한 판단 뿐 아니라 객체들 간의 관계를 파악하고 자연어의 형태로 알맞게 표현하는 문제를 포함하고 있어 컴퓨터 비전 기술과 자연어 처리 기술이 결합되어있다. 이미지 캡셔닝은 이미지 데이터를 텍스트로 바꾸어주기 때문에 데이터의 양을 획기적으로 줄일 뿐 만 아니라, 데이터에 대한 접근성도 용이하게 해준다. 앞서 말한 이미지, 영상 데이터에서 필요한 내용을 검색하는 예시를 포함하여 다양한 분야에서 사용된다. 시각 장애인에게 상황 설명을 음성으로 제공, 청각장애 아동의 학습 교재 등으로 사용, 가정용/보안용 CCTV 영상을 텍스트로 변환, 영상 컨텐츠 줄거리 요약 등 일상생활에서 유용하게 활용될 수 있는 기술이다.

이미지 캡셔닝 기술은 이미 여러 연구를 통해 좋은 성능을 보이고 있지만, 대다수의 공개된 이미지 캡션 데이터는 영어로 이루어져 있어 한국어 캡션을 생성하기 위해서는 영어로 생성한 캡션을 한국어로 번역해서 사용해야 했다. 이 연구에서는 한국어 이미지 캡션 모델을 구성하여 영어 이미지 캡션 모델을 번역하여 한국어 이미지 캡션을 만드는 것과 한국어 이미지 캡션을 곧바로 생성하는 모델의 성능을 비교한다.

2. 모델 및 방법

본 연구에서 사용한 이미지 캡션 모델의 구조는 Show, Attend and Tell[1]연구를 기반으로 한다. 이 모델은 이미지를 해석하는 인코더 부분, 이미지를 해석한 데이터로 캡션을 생성하는 디코더 부분이 결합된 형태[2]에서 디코더에 Attention Mechanism을 추가한 모델이다. 모델의 개요는 그림 1에 나타내었다. 인코더는 이미지를 Convolutional Neural Network(CNN)[3]를 거쳐 Feature Map을 생성한다. CNN 모델은 사전 학습된 모델을 사용하여 좋은 성능을 유지하게끔 한다. 사전 학습 되어있는 CNN 모델로는 ResNet-101[4] 모델을 사용하였다. 기존의 ResNet 모델은 이미지를 분류하기 위해 모델의 말단에 Fully-Connected Layer를 통한 Classification이 일어나지만, 여기서는 이미지의 feature vector만 추출해 사용하므로 Fully-Connected Layer를 제거한 후 사용한다. 이미지가 인코더에 입력되면 ResNet-101 모델을 거치며 2048차원의 14*14 feature map이 생성된다. 인코더를 통해 출력되는 형태는 (2048, 14, 14) 크기의 텐서이다.

디코더는 인코딩된 이미지로부터 한 단어씩 캡션을 생성한다. 시퀀스를 생성하기 위해 LSTM을 사용한다. 일반적인 LSTM에서는 단순히

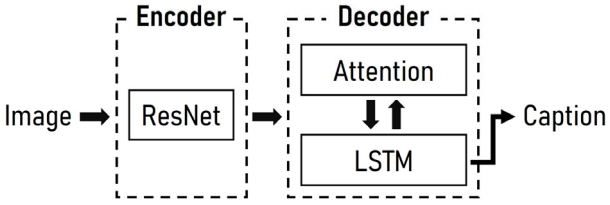

그림 1. Show, Attend and Tell 모델 구조[1]

인코딩된 이미지의 모든 픽셀에 걸쳐 디코더를 거쳐 다음 단어를 생성하고, 생성된 단어를 통해 다음 단어를 새로 생성한다. 이 LSTM에 Attention Mechanism을 적용하여 이미지의 어느 부분에 집중하여 다음 단어를 생성할지 결정하도록 한다. Attention Mechanism에는 Hard-Attention과 Soft-Attention이 있고, 이 연구에서는 Soft-Attention을 사용하였다.

3. 실험 데이터

학습 데이터는 MS-COCO 2014 데이터셋[5]을 사용하였다. MS-COCO 이미지 데이터셋에는 총 123,287장의 이미지가 있고, 각각의 이미지는 5개의 캡션을 가지고 있다. 한국어 캡션 생성 모델의 성능을 평가하기 위해, 같은 환경에서 영어 캡션 생성 모델과 한국어 캡션 생성 모델을 학습하였다. 영어 캡션 생성 모델 학습에는 영어 캡션 데이터셋을 사용하였고, 한국어 캡션 생성 모델 학습에는 한국어 캡션 데이터셋을 사용하였다. 한국어 캡션은 MS-COCO 캡션 데이터셋[6]을 AI Hub가 제공하는 기계번역 후 오류 수정한 데이터셋[7]을 사용하였다. 같은 이미지 데이터셋과 같은 내용의 캡션 데이터셋을 사용함으로써 언어 차이로 발생하는 모델의 특징을 비교적 정확하게 파악할 수 있다.

한국어 캡션 데이터셋은 문장 단위로 구성되어있다. 캡션을 학습시키는 과정은 문장 안의 단어 간 의미 파악이 포함되어있다. 이를 위해 캡션을 토큰 단위로 나누어 사용하는 전처리 과정을 거친다. 영어의 경우 토큰은 띄어쓰기 단위로 나뉘지만, 한국어는 어미와 조사의 변화로 인해 기존 방식인 어절 단위로 토큰화 하는데 어려움이 있다. 형태소 단위로 토큰을 생성하기 위해 한국어 정보처리를 위한 파이썬 패키지인 KoNLPy[8]의 형태소 분석 엔진 Mecab 라이브러리를 사용하여 토큰화 하였다. Mecab은 텍스트를 입력하면 이를 형태소 단위와 형태소 태그를 출력한다. Mecab을 이용한 형태소 분석 예시는 그림 2에 나타내었다. 이를 이용해 한국어 캡션 데이터셋을 형태소 단위로 변형하여 전처리 과정을 거친 후 네트워크 모델에 학습하였다.

Sentence
: 한 남자가 자전거를 타고 있다.

Tokenized
: [('한', 'MM'), ('남자', 'NNG'), ('가', 'JKS'), ('자전거', 'NNG'), ('를', 'JKO'), ('타', 'VV'), ('고', 'EC'), ('있', 'VX'), ('다', 'EF'), ('.', 'SF')]

그림 2. 한국어 문장의 토큰화 예시

4. 실험 및 결과

이 연구에서 모델 학습을 위해 CPU Intel Core i7-4790, GPU NVIDIA의 Geforce GTX 2080 Ti, RAM 32GB의 시스템에서 Pytorch 라이브러리로 실험하였다. 모델 학습을 위해 총 123,287장의 데이터 중 113,287장을 학습 데이터셋으로 사용하고 5,000장으로 검증하였다. 남은 5,000장을 테스트셋으로 사용하였다. 영어 캡션 생성 모델과 한국어 캡션 생성 모델 모두 인코더는 사전학습 되어있는 그대로 사용하였고 디코더만 학습하였다.

한국어 캡션을 생성하는 과정에서, 디코더는 순차적으로 토큰을 출력한다. 한국어에서의 토큰은 형태소이므로, 이를 최종 출력하면 형태소 단위로 띄어쓰기가 되는 것을 확인할 수 있다. 이를 해결하기 위해, Mecab 형태소 분석기를 다시 활용하였다. 한국어 캡션을 띄어쓰기가 없는 상태로 출력한 후, 이를 Mecab으로 형태소 분석을 하였다. 이후 띄어쓰기 방법에 맞게 공백을 추가하여 최종 한국어 캡션을 출력하였다.

그림 3에서는 테스트 데이터셋의 이미지로부터 영어, 한국어, 영어를 번역한 한국어 캡션을 생성하여 이미지의 reference 캡션과 성능을 비교하였다. 영어로 생성한 캡션을 한국어로 번역하는 과정에서는 Google Cloud Translate API[9]를 사용하였다. 각각의 이미지는 5가지 캡션을 보여주고 있다. 'True(en)'은 영어 Ground Truth 캡션, 'True(ko)'는 영어 Ground Truth 캡션을 번역한 한국어 Ground Truth 캡션이다. 'en'은 영어 생성 캡션, 'ko'는 한국어 생성 캡션, 'en-ko'는 영어 생성 캡션을 한국어로 번역한 캡션을 뜻한다. 그림 3의 (a), (b) 에서는 Ground Truth 캡션과 새로 생성한 캡션이 일맥상통하는 것을 확인할 수 있다. 다만 그림 3의 (c)에서는 한국어 캡션에서는 '주차된'으로 표현된 영어 단어 'sitting'이 한국어로 번역되며 '앉아 있는'으로 문맥에 맞지 않게 번역된 것을 확인할 수 있다. 그림 3의 (d)에서는 생성한 캡션이 이미지의 장소를 잘못 예측한 것을 확인할 수 있다.

모델의 성능 평가 방법으로는 기계번역에서 성능 지표로 사용하는 BLEU 스코어를 사용하였다. BLEU 스코어는 n-gram 방식으로 n개의 연속된 단어 개수씩 비교하여 문장의 일치 정도를 평가한다. BLEU-1은 하나의 단어 단위, BLEU-4는 4개의 단어 단위로 문장을 비교하며, 전체 문장의 일치 정도를 평균으로 하여 최종 결과 값을 산출한다. 또한 캡션을 생성하는 과정에서 Beam Search[10]를 적용하여 Beam Size별로 BLEU 스코어를 산출하였다. 그림 3은 Beam Size 3으로 지정하여 캡션을 생성하였다.

영어 캡션 생성 모델과 한국어 캡션 생성 모델을 평가하여 표 1과 표 2에 결과를 나타내었다. 두 모델의 BLEU 스코어 측정 결과, Beam Size 3에서 BLEU-4 스코어는 영어 캡션 생성 모델에서 30.57, 한국어 캡션 생성 모델에서 41.67이 나왔다. 한국어 캡션 생성 모델이 영어 캡션 생성 모델에 비해 전체적으로 높은 스코어를 갖는다. 하지만 이는 언어의 차이로 인한 BLEU 스코어의 차이로 해석할 수 있다.

이를 확인하기 위해 영어로 생성된 이미지 캡션을 한국어로 번역하여 생성한 한국어 캡션으로 BLEU 스코어를 측정하였다. 결과는 표 3에 나타내었다. 영어 캡션을 한국어로 번역하여 BLEU 스코어를 측정한 결과, Beam Size 3에서 BLEU-4 스코어는 25.48이 나왔다. 한국어 캡션 생성 모델을 평가한 결과가 영어 캡션을 한국어로 번역하여 평가한 것보다 전체적으로 좋은 스코어를 보인다. 이를 통해 한국어 캡션 생성 모델이 기존의 영어 캡션 생성 모델을 번역하는 것 보다 좋은 성능을 보이는 것을 확인하였다.

True(en)	A dog that is in the back of a car.
True(ko)	차 뒤에 있는 개.
en	a dog sitting in the back seat of a car
en-ko	자동차 뒷좌석에 앉아있는 개
ko	개가 차안에 앉아 있다.

(a)

True(en)	A bus is going down the road at night.
True(ko)	밤에 버스가 길을 따라 내려가고 있다.
en	a city street filled with lots of traffic
en-ko	교통량이 많은 도시 거리
ko	밤에 시내거리를 달리는 버스

(b)

True(en)	The plane is parked at the gate at the airport terminal.
True(ko)	비행기가 공항 터미널 정문에 주차되어 있다.
en	a white airplane sitting on top of an airport runway
en-ko	공항 활주로 위에 앉아 있는 흰색 비행기
ko	공항활주로에 주차된 비행기

(c)

True(en)	A classroom that is empty and has tables and chairs.
True(ko)	비어 있고 테이블과 의자가 있는 교실
en	a living room filled with lots of furniture
en-ko	많은 가구로 가득 찬 거실
ko	식탁과 의자가 있는 부엌

(d)

그림 3. 이미지 캡션 생성 예시

Beam size	BLEU-1	BLEU-2	BLEU-3	BLEU-4
1	70.22	52.54	37.97	27.36
3	71.36	54.20	40.57	30.57
5	71.02	54.03	40.62	30.86

표 1. 영어 캡션 모델 평가

Beam size	BLEU-1	BLEU-2	BLEU-3	BLEU-4
1	74.86	60.77	48.89	38.58
3	75.55	62.48	51.39	41.67
5	74.73	62.07	51.34	41.97

표 2. 한국어 캡션 모델 평가

Beam size	BLEU-1	BLEU-2	BLEU-3	BLEU-4
1	60.94	45.12	33.40	24.86
3	59.01	44.15	33.32	25.48
5	58.58	43.93	33.32	25.61

표 3. 영어 캡션을 번역한 한국어 캡션 평가

5. 결론

이 연구에서는 기존의 이미지 캡션 생성 모델을 이용해 한국어 캡션을 직접 생성하는 모델을 만들고자 하였다. 한국어 캡션을 생성하는 방법으로는 기존의 영어 캡션 생성 모델을 이용해 영어로 캡션을 생성하여 한국어로 번역하거나 한국어 캡션을 직접 생성하는 방법이 있으며, 두 캡션 생성 방법의 성능을 직접 비교하였다. 한국어 캡션 생성 모델의 성능을 기존 영어 캡션 생성 모델의 성능과 비교하기 위해, 영어 캡션 학습 데이터셋을 한국어로 번역한 데이터셋을 활용하여 같은 조건에서 같은 내용의 캡션으로 영어 캡션 생성 모델과 한국어 캡션 생성 모델을 각각 학습하였다. 영어 캡션 생성 모델과 한국어 캡션 생성 모델의 BLEU 스코어를 측정하였고, Beam Size 3에서 BLEU-4 스코어는 영어 캡션 생성 모델에서 30.57, 한국어 캡션 생성 모델에서 41.67이 나왔다. 이 차이는 영어와 한국어의 차이를 포함할 수 있기 때문에, 영어 캡션 모델이 생성한 캡션을 한국어로 번역하여 측정하여 결과를 비교하였다. 그 결과 영어로 생성한 캡션을 한국어로 번역하여 평가하니 Beam Size 3에서 BLEU-4는 25.48이 측정되었다. 이를 통해 한국어로 이미지 캡션을 생성하는 모델을 목적에 맞게 생성한 것을 확인하였다.

감사의 글

본 논문은 과학기술정보통신부의 재원으로 IITP (NO. 2020-0-00994, 이용 환경을 반영하는 자율적 VR・AR 콘텐츠 생성 기술개발)의 지원을 받아 수행되었음.

참고문헌

[1] K. Xu, J. L. Ba, R. Kiros, K. H. Cho, A. Courville, R. Salakhutdinov, R. Zemel and Y. Bengio, “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention,” Computer Vision and Pattern Recognition, Proceedings of the 32nd International Conference on Machine Learning, PMLR 37:2048-2057, 2015.

[2] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, “Show and Tell: A Neural Image Caption Generator,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3156-3164

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Commun. ACM 60, 6 (June 2017), 84-90.

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778

[5] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár & C. L. Zitnick, “Microsoft coco: Common objects in context.” European Conference on Computer Vision. Springer International Publishing, 2014.

[6] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick, "Microsoft COCO captions: Data collection and evaluation server." arXiv preprint arXiv:1504.00325 (2015).

[7] https://aihub.or.kr/aihubdata/data/view.do?currMenu=120&topMenu=100&aihubDataSe=extrldata&dataSetSn=261

[8] E. J. Park and S. Z. Cho, “KoNLPy: Korean natural language processing in Python,” Proceedings of the 26th Annual Conference on Human and Cognitive Language Technology, Gangwon-do, Korea, pp. 133-136, 2014.

[9] https://cloud.google.com/translate

[10] R. Socher, C. C. Lin, A. Y. Ng, and C. D. Manning, “Parsing natural scenes and natural language with recursive neural networks.” In Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML'11). Omnipress, Madison, WI, USA, 129-136. 2011.