

Dive to Visual Question Answering

Korea University COSE461 Final Project

이창수
Department of Computer Science
Team 12
2016320104

강인구
Department of Computer Science
Team 12
2016320115

예승형
Department of Economics
Team 12
2017150358

정경륜
Department of Computer Science
Team 12
2018320142

Abstract

본 연구는 VQA task에서 어떤 부분을 변경하였을 때 정확도가 향상되는지를 살펴보았고 최종적으로 VQA 모델의 정확도를 높이기 위한 시도를 하였다. Vanilla VQA에서 시작해 모델의 구조를 변형시키며 나아갔다. 첫 번째론 기존의 vanilla VQA에서 언어 모델을 LSTM 기반 모델에서 SBERT 기반 모델로 바꾸는 시도를 하였다. 이 경우 학습이 LSTM보다 좋은 수준으로 빠르게 수렴하지만, 극적인 변화는 보여주지 못했다. 이미지 채널과 텍스트 채널을 독립적으로 학습시킬 경우 한계가 있음을 확인했고, cross-modal attention 기법을 적용하기로 결정했다. cross-modal attention 기법을 통해 VQA의 성능을 크게 개선한 MCAoAN 모델 구조를 분석하고 이를 바탕으로 모델링을 하여 학습시켰다. 또한 attention 시각화를 통해 MCAoAN 기반 모델이 주어진 문제에 대해 이미지와 자연어에서 의미 있는 정보를 집중적으로 보고 있다는 것을 확인할 수 있었다. 이 모델을 기반으로 한 demo는 <http://vqateam12.kro.kr/>에서 확인해볼 수 있다. 이후 VQA와 유사한 task인 Visual Grounding을 multi task learning 기법을 통해 학습시켜서 성능 향상을 꾀해보았지만, Visual Grounding 모델과 VQA 모델이 성능 차이가 크게 나서 효과를 보지 못했다. 모델의 Visual Grounding decoder 레이어의 고도화는 term project의 시간 문제로 발전시키지 못했다. 구현 코드는 https://github.com/leecs0503/NLP_FinalProject에서 확인해볼 수 있다.

1 Introduction

Visual Question Answering (VQA)은 이미지와 질문이 동시에 주어졌을 때 질문에 대한 올바른 답을 도출하는 것을 목표로 하는 연구이다. 이 연구가 2015년에 등장하면서 기존의 QA 시스템에서 이미지나 자연어 둘 중 하나만 보고 답을 도출하는 것에서 벗어나 인간의 사고와 비슷하게 시각적, 언어적 정보를 활용해 질문을 이해하는 시스템으로 발전될 수 있었다. 첫 VQA 연구 발표 이후 언어 정보와 시각 정보를 통합하기 위해 다양한 변형 모델이 나왔고 이 정보들을 어떻게 조합하고 처리하는지에 따라 다양한 결과가 도출되었다. 이렇게 서로 다른 영역의 정보를 통합해서 처리하는 방법론의 발전을 통해 인공지능이 인간의 사고방식에 가까워질 수 있기 때문에 향후 딥러닝의 발전에 있어서 중요한 주제가 될 것이다. 따라서 각 도메인들을 적절히 결합해 성능을 개선해 볼 필요가 있다.

VQA 모델은 크게 이미지 처리, 자연어 처리, 이미지와 자연어 처리의 결합 세 가지 모듈로 구성되어 있다. Vanilla VQA

에선 이미지 처리는 VGG19를, 자연어 처리는 LSTM을 이용하고 두 정보의 결합은 element-wise multiplication을 통해 이루어졌다. 본 연구에선 이 vanilla VQA에서 시작해 위에서 언급한 3가지 모듈의 변형을 통해 어떤 VQA 구조에서 성능의 발전을 꾀할 수 있는지 살펴보려고 한다. 또한 실험 결과에 대한 시각화를 통해 VQA 모델이 어떻게 이미지와 자연어 구조를 이해하고 있는지에 대해서 분석하였다.

2 Related Work

VanillaVQA[1]는 VQA 연구의 시초가 된 논문으로 이미지와 자연어를 하나로 결합하는 모델을 소개하고 VQA를 위한 데이터 세트와 모델 평가 방법을 정의했다. 본 연구도 이 vanilla VQA의 baseline에서 시작한다.

LSTM[2]은 RNN의 hidden state에 cell-state를 추가함으로써 RNN의 vanishing gradient problem을 해결한 모델이다. 본 연구에서는 VQA의 자연어 모델의 기본 구조로 사용하였다.

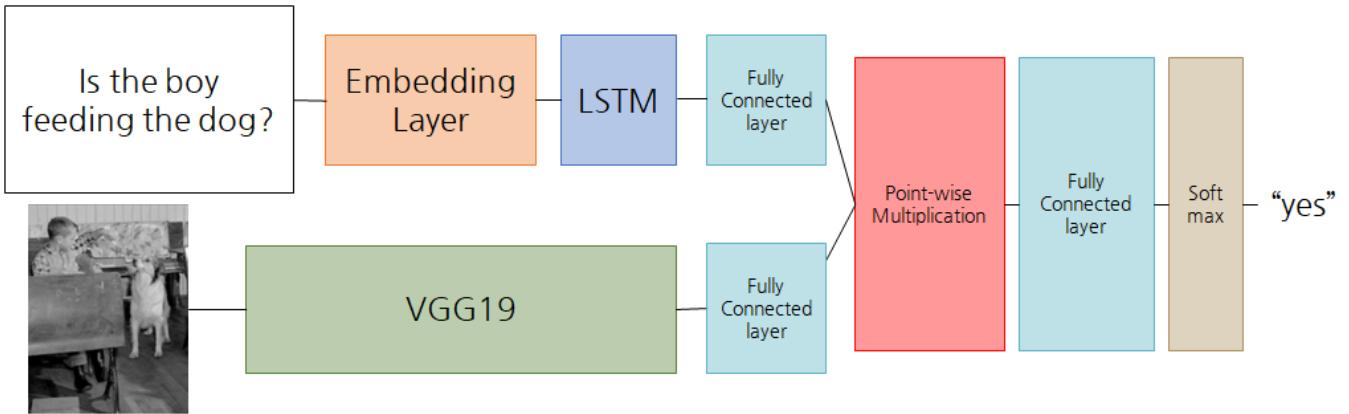


Figure 1: LSTM based VQA architecture

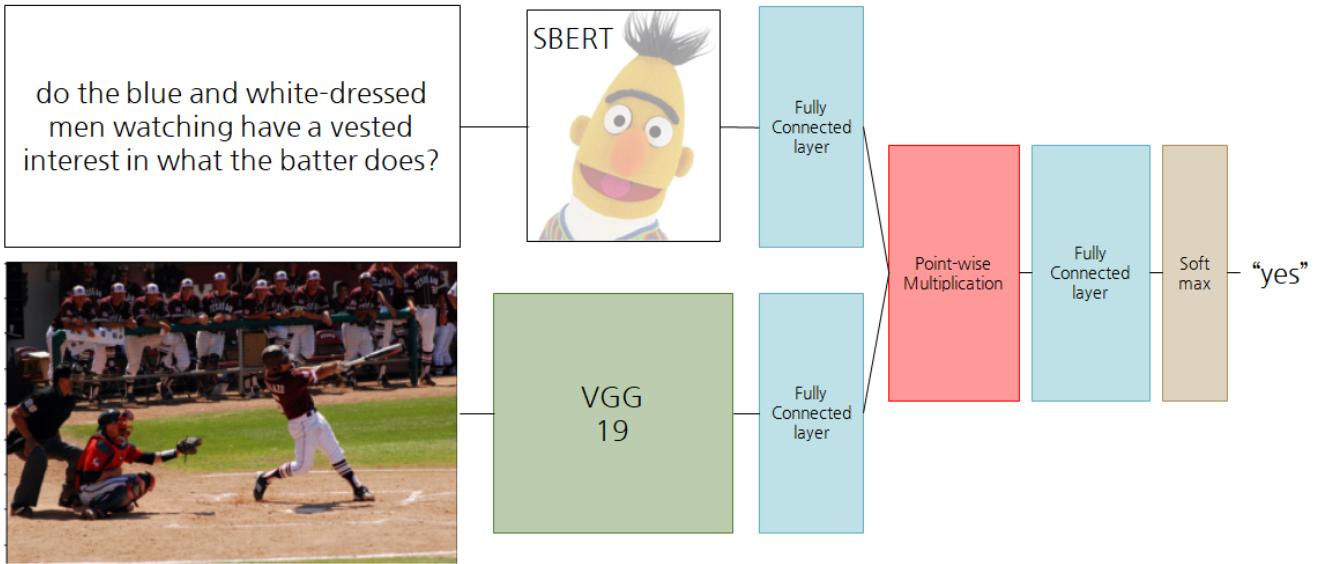


Figure 2: SBERT architecture

VGGNet[3]은 CNN의 depth를 늘리면서도 vanishing gradient 문제를 해결한 연구이다. 본 연구에선 layer를 19개 사용한 VGG19를 사용하였으며, VQA 모델에서 이미지를 처리하는 모델의 기본 구조로 사용하였다.

BERT[4]는 transformer의 encoder를 bidirectional한 방향으로 적용한 연구로 가장 큰 특징이라면 대용량의 unlabeled data를 이용해 pretrain된 모델을 제공한다는 점이다. 이를 이용해 전이학습을 시행할 수 있다. 본 연구에선 VQA의 언어 모델 중 하나로 BERT를 활용했다.

SBERT[5]는 BERT로 부터 문장의 유사도를 구하는 문제를 통해 pooling layer을 학습시킴으로써 문장의 임베딩을 얻을 수 있는 모델을 제시한다. 본 연구에서 vanilla VQA를 개선하는 과정에서 질문의 문장 벡터를 얻어 임베딩을 진행하는 연구를 진행하였다.

Transformer구조[6]는 기존의 RNN기반의 구조보다 다양한 분야에서 성능 향상을 보여주었다. 본 연구에서 학습시킨 MCAoAN 모델에서 해당 구조를 채택하고 있다.

Faster R-CNN[7]은 Region Proposal Network(RPN)를 통해서 RoI를 계산하는 방법을 제시한 논문이다. 본 연구에서 이미지 정보의 처리를 개선하기 위해 위 논문을 참조하였다.

Bottom up attention[8]은 faster-rcnn을 이용하여 실제 VQA에서 활용할 RoI feature embedding을 뽑는 방법을 제시하였다.

MCAN[9]은 bottom-up attention을 이용한 image feature와 LSTM을 통한 word embedding을 각각 transformer구조를 통해 기존 VQA 모델에서 이미지 RoI를 적용시켜 개선시킨 모델이다.

MCAoAN [10]은 MCAN을 개선시킨 모델로 Attention on Attention 기법과 multi-modal attention fusion을 통해 성능을 향

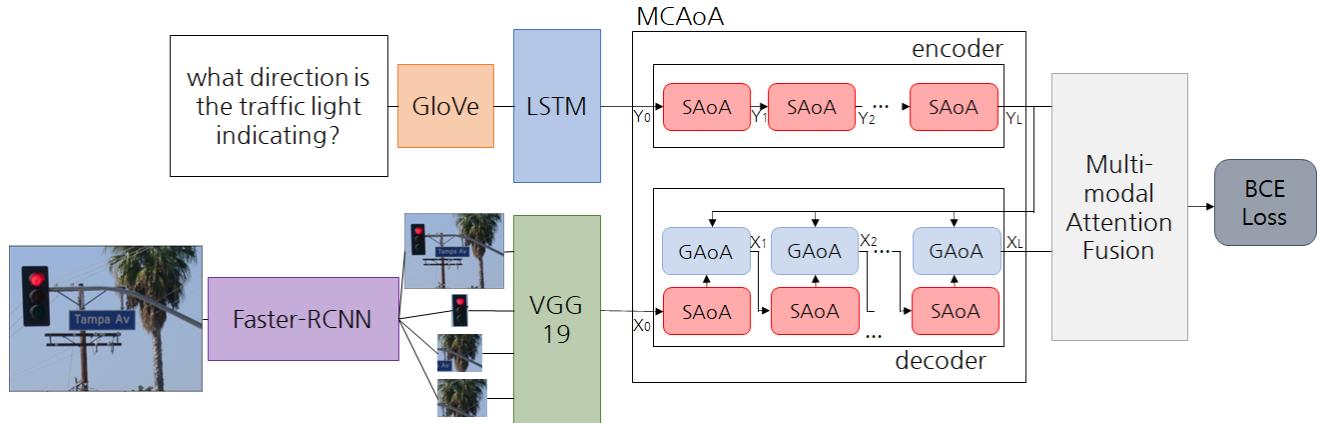


Figure 3: MCAoAN architecture

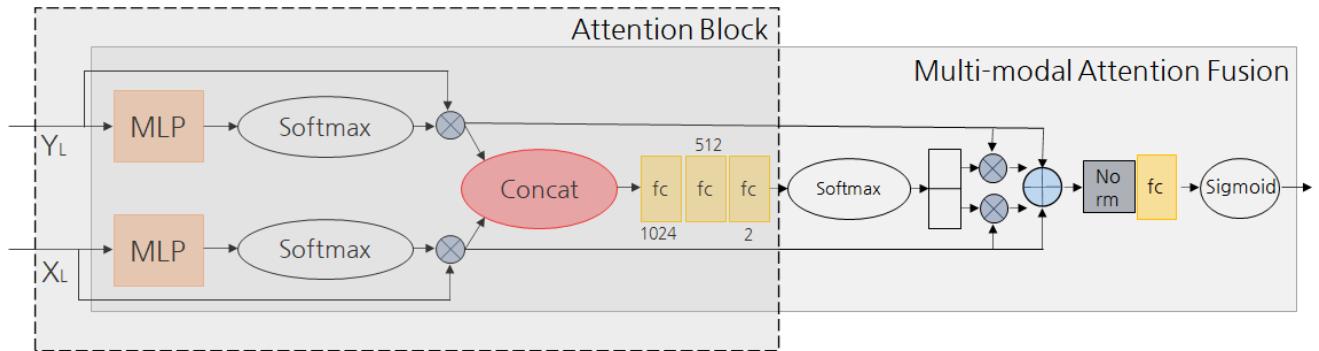


Figure 4: Multi-modal Attention Fusion

상시켰다. 본 연구에서는 해당 모델에 대한 구현을 진행하였고, 각 데이터에 대해 attention 시각화를 진행해 분석하였다.

3 Model

본 연구에선 vanilla VQA를 기반으로 하는 LSTM + VGG19 based VQA에서 시작해 이미지 모델과 자연어 모델, 이미지와 자연어의 결합부를 다양한 방식으로 변형을 시도하면서 VQA가 어떻게 이미지와 자연어 정보를 이해하고 처리하는지에 대해서 직접 구현하면서 탐구하였다. 구현한 VQA 모델은 크게 3가지로 baseline이 되는 LSTM + VGG19 based VQA, LSTM based VQA에서 언어모델을 LSTM 대신 SBERT로 대체한 SBERT + VGG19 based VQA, 그리고 attention을 기반으로 한 MCAoAN VQA가 있다.

3.1 LSTM + VGG19 based VQA

우선 LSTM based VQA의 구조는 Figure 1에서 볼 수 있다. LSTM based VQA는 vanilla VQA의 모델 구조를 그대로 차용하였다. 이미지 모델은 pretrained VGG19를 적용하였다. VGG19를 거치고 난 이후 l2 normalize를 적용하였다. 자연어 모델은 LSTM을 기본 구조로 사용하였다. LSTM의 last hid-

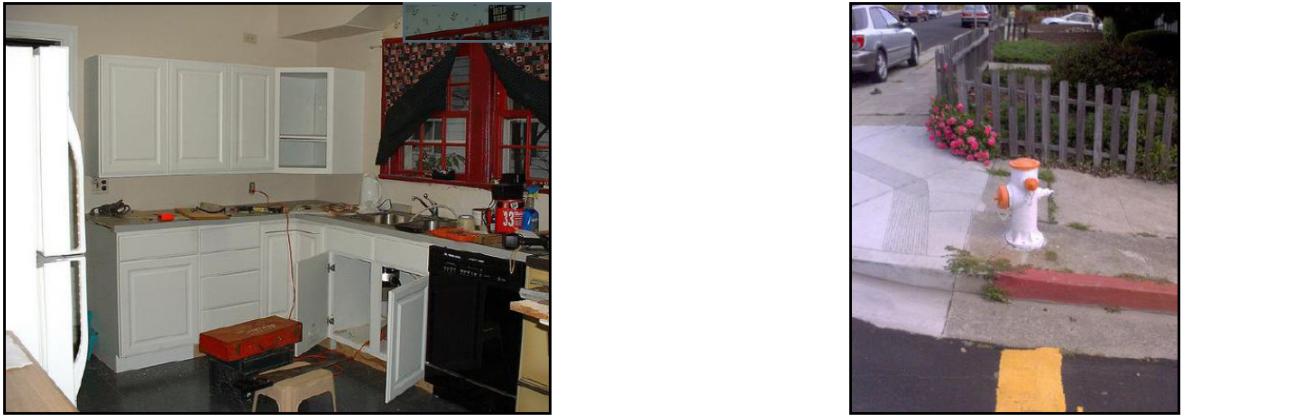
den state는 512-dim을 적용하였고 이 LSTM을 통해 question에 대한 1024-dim의 embedding vector를 얻어낸다. 각 모델에서 추출한 embedding vector를 결합하기 위해서 두 vector를 point-wise multiplication을 취한다. 이후 fully-connected layer와 softmax를 거쳐 question에 대한 answer를 추론한다.

3.2 SBERT + VGG19 based VQA

SBERT VQA 구조는 Figure 2에서 볼 수 있다. LSTM 기반 VQA는 길게 구성된 질문의 embedding을 제대로 구하지 못하고, 실제 현실에서의 robust한 질문을 처리하지 못하기 때문에 BERT를 사용해 transfer learning 기법을 활용해보기로 결정했다. 질의 전체의 embedding을 구하기 위해 두 문장의 유사도를 비교함으로써 pooling layer를 학습하는 Sentence-Bert를 활용했다. 다양한 도메인에서의 56만개의 문장을 통해 문장을 768-dim embedding vector로 변환시켜주는 pretrain 된 SBERT모델을 LSTM대신 활용하였다. 그 외 다른 구조는 위의 LSTM + VGG19 based VQA와 동일하다.

3.3 MCAoAN based VQA

마지막으로 MCAoAN 기반 VQA 구조는 Figure 3에서 볼 수 있다. 우선 자연어 모델은 최대 길이가 n인 question을 GloVe



Is something under the sink broken?	yes yes no yes	no	no
What number do you see?	33 33 33	5 6 7	
Can you park here?		no no no	no no yes
What color is the hydrant?		white and orange white and orange white and orange	red red yellow

Figure 5: VQA data example

를 통해 임베딩을 한 후 LSTM에 넣어 값을 추출한다. 그리고 이미지 모델은 faster-RCNN을 통해서 m개의 ROI를 뽑아낸 후 이를 다시 VGGNet에 넣는다. 그 후 각 모델에서 추출한 값을 MCAoAN에 넣게 된다. MCAoAN은 L(=6)개 층의 SAoA로 이루어진 encoder와 SAoA + GAoA로 이루어진 decoder로 이루어져 있다. SAoA(Self Attention on Attention)와 GAoA(Guided Attention on Attention)은 AoA(Attention on Attention)로 이루어진 layer으로, AoA block + Feed forward network + dropout(0.1) + layernorm으로 이루어져 있다. SAoA는 Key, Value, Query가 전부 같은 self attention이고, GAoA는 Key, Value는 encoder를 통과한 attention, Query는 decoder의 input인 attention이다. AoA block은 scaled dot multi-head attention (f_{att})에서 추가적으로 구해진 attention과 query 간의 relation을 계산하는 층으로 Information Gate(I)와 Attention Gate(G)의 elementwise multiplication을 통해 구하는데 그 식은 다음과 같다.

$$f_{att} = \text{Softmax}\left(\frac{QK}{\sqrt{d}}\right)V$$

$$I = W_Q Q + W_{Q_{att}} f_{att} + b_I$$

$$G = \sigma(W_G Q + W_{G_{att}} f_{att} + b_G)$$

$$W_Q, W_{Q_{att}}, W_G, W_{G_{att}} \in R^{d \times d}$$

$$b_I, b_G \in R^d$$

구해진 각각의 $R^{m \times d}$ 의 image feature X와 $R^{n \times d}$ 의 text feature Y는 multi-modal attention fusion(Figure 4)을 통해 정답의 distribution을 구한다. n개의 text feature vector와 m개의 image feature vector는 weighted sum을 통해 R^d 로 압축되는데, MLP layer(FC(d) - ReLU - dropout(0.1) - FC(1))와 softmax

를 통해 weight가 계산되고, image와 text에 대한 attended feature가 계산되는데 그 식은 다음과 같다.

$$X' = \sum_{i=1}^m \text{Softmax}(MLP(X))x_i$$

$$Y' = \sum_{i=1}^n \text{Softmax}(MLP(Y))y_i$$

이 두 feature 또한 weighted sum을 통해 combined feature로 합쳐지고, 이를 위해 각 feature의 concatenation을 한 뒤 여러 fc layer(FC(2d) - dropout(0.2) - FC(d) - dropout(0.2) - FC(2) - softmax)를 통과시켜 weight가 계산된다. 이렇게 구해진 combined feature에는 layernorm - fc layer - sigmoid activation을 통해 최종 distribution이 계산된다.

4 Experiments

4.1 Data

VQA 연구를 위한 데이터 세트은 <https://visualqa.org>에서 제공하는 VQA v2 데이터셋을 그대로 활용하였다. VQA v2 데이터 세트은 real image와 abstract image, question과 answer 데이터를 제공한다. 그 중 실험의 편의성을 위해 abstract image는 제외하고 204,721 장의 real image와 1,135,904 개의 question 데이터, 6,581,110의 answer 데이터만을 가져와 사용하였다. VQA 데이터셋의 실제 예시는 Figure 5에서 볼 수 있다.

Question 데이터는 상식만으로는 대답할 수 없고 이미지를 보고 답변을 할 수 있는 다양한 질문들로 구성되어 있다. Question의 유형은 크게 open-ended, 다지선다형(객관식) 등이 있다. Answer 데이터는 question의 유형별로 적절한 answer가 제공된다. Open-ended 질문에 대한 answer의 유형은

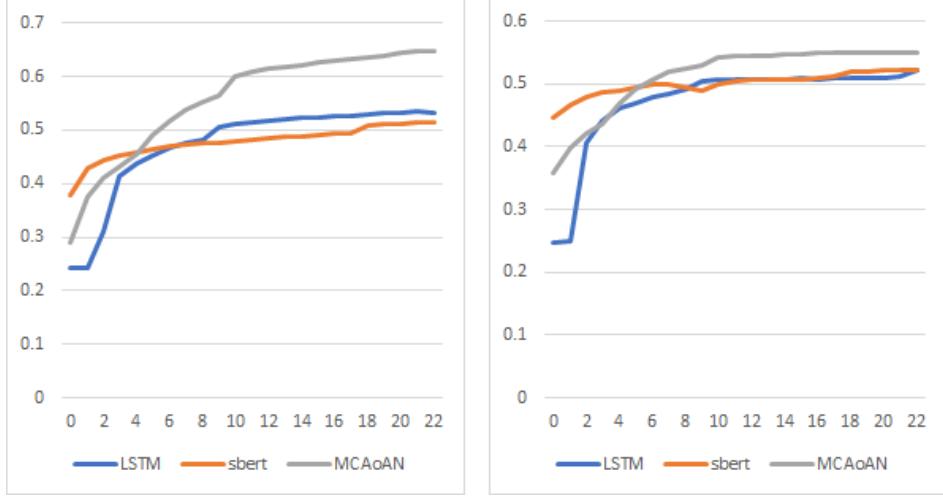


Figure 6: Left : Training Accuracy | Right : Validation Accuracy

대부분 yes/no로 구성되어 있으며 그 외 숫자나 짧은 단어 등이 있다. 다지선다형 문제는 18개의 선택지가 존재한다.

4.2 Evaluation method

모델 평가 방법은 각 모델의 accuracy를 측정하여 비교한다. VQA-v2 test dataset에 annotation이 따로 적혀있지 않기 때문에 VQA-v2 validation dataset을 절반으로 나눠 하나는 validation에 활용하고 다른 하나는 모델 accuracy를 측정하기 위한 test에 활용한다.

Open-ended 질문들은 다음 metric으로 accuracy를 측정한다.

$$\bullet \text{accuracy} = \min\left(\frac{\text{Number of right answer}}{3}, 1\right)$$

또한 자체 개발한 데모 웹사이트를 이용해 각 VQA 모델들이 실제로 얼마나 정확하게 VQA task를 수행하고 있는지 확인해 보았다.

4.3 Experimental details

각 모델은 다음과 같은 조건에서 실행되었다.

4.3.1 LSTM + VGG19 based VQA

해당 모델의 Configurations로는 word embedding size = 300, output embedded size = 1024, hidden state size of LSTM = 512, number of stacked LSTM = 2로 구성이 되어 있다. Hyperparameter로 dropout rate = 0.5를 사용하였고 optimizer로는 adam solver를 사용하였으며 $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate = 0.001으로 실행되었으며 learning rate decay는 10epoch마다 1/10만큼 감소시켰다. Batch size는 256으로 총 25epoch까지 학습을 진행하였고, 17epoch까지 진행한 모델을 선택했다.

4.3.2 SBERT + VGG19 based VQA

해당 모델의 Configurations로는 output embedded size = 1024로 구성이 되어 있다. Hyperparameter는 dropout rate = 0.5

를 사용하였고 optimizer로는 adam solver를 사용하였으며 $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate = 0.001으로 실행되었으며 learning rate decay는 10epoch마다 1/10만큼 감소시켰다. Batch size는 256으로 총 25epoch까지 학습을 진행하였고, 19epoch까지 진행한 모델을 선택했다.

4.3.3 MCAoAN based VQA

해당 모델의 Configurations로는 resnet50을 backbone으로 가지며 COCO train2017로 pretrain된 faster rcnn 모델, 이미지 최대 갯수=20 embed size=64, AoA block의 multi head의 갯수=8로 구성이 되어 있다. optimizer로는 adam solver를 사용하였으며 $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate = 0.001으로 실행되었으며 learning rate decay는 10epoch마다 1/10만큼 감소시켰다. Batch size는 256으로 총 25epoch까지 학습을 진행하였고, 13epoch까지 진행한 모델을 선택했다.

4.4 Results

Figure 6을 살펴보면 LSTM과 SBERT는 MCAoAN에 비해 낮은 성능을 보이고 있음을 알 수 있다. LSTM은 2epoch까지 accuracy가 가파르게 증가하다가 천천히 수렴하는 경향을 보이고 있고, 반면 SBERT는 LSTM에 비해 훨씬 빠른 학습 속도를 보이지만 둘 다 accuracy가 0.5에서 수렴하고 있다. MCAoAN은 LSTM과 SBERT보다 train data에 과적합되는 경향이 나타난다.

Acc	LSTM	SBERT	MCAoAN
all(val)	0.5213	0.5222	0.5496
all(test)	0.5044	0.5143	0.5573
yes/no	0.6928	0.7123	0.7391
num	0.3870	0.3844	0.4310
others	0.3892	0.3901	0.4477

test set을 이용하여 테스트한 결과 전반적으로 yes/no에 대한 질문에 대한 답을 다른 분야에 비해 더 잘하는 것으로 나타났다. Vanilla VQA의 LSTM을 SBERT로 개선한 모델은 다른 질문에 비해 yes/no 질문에 대한 정확도가 개선되었다는 것

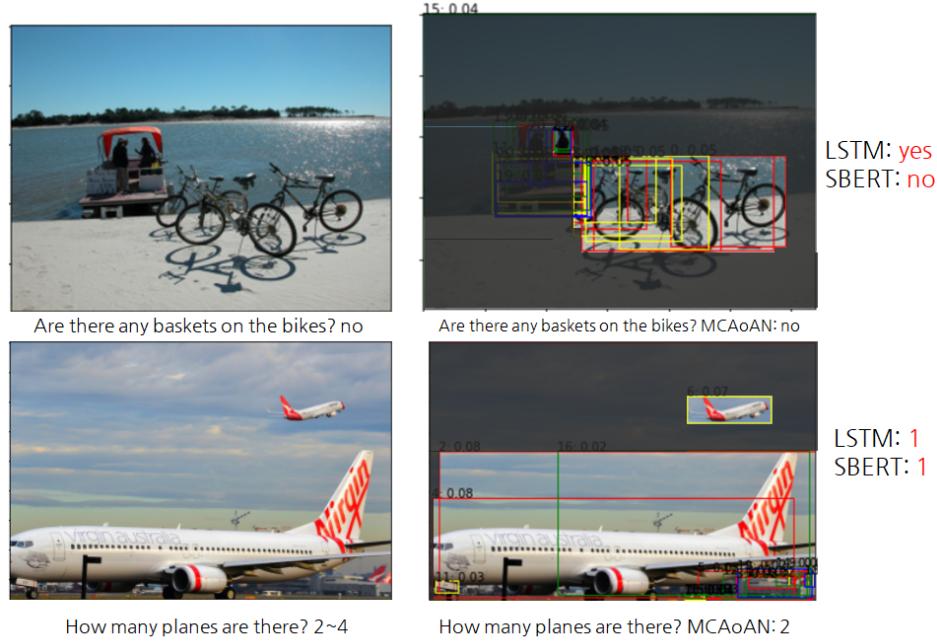


Figure 7: MCAoAN이 잘 대답한 질문들과 그와 관련된 attention

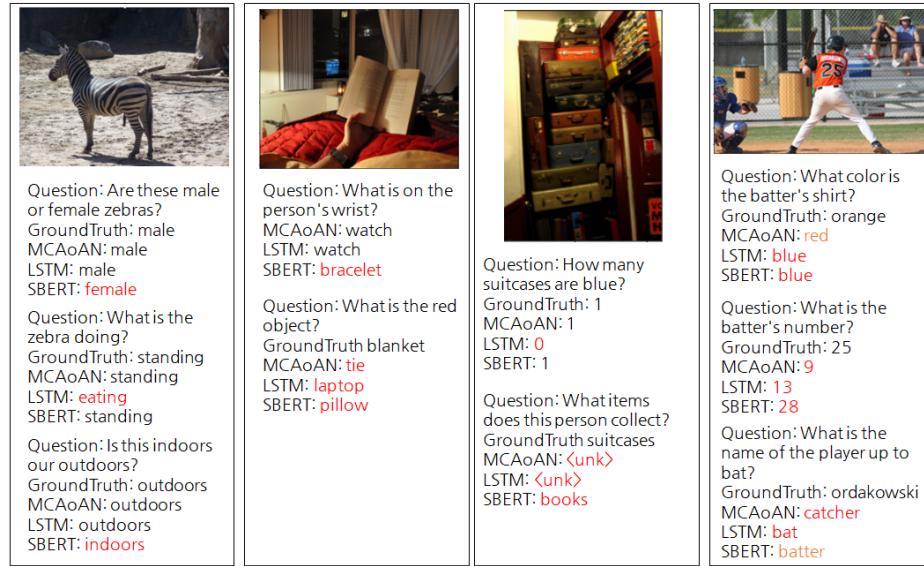


Figure 8: 각 모델들이 제대로 대답하지 못한 질문들 (빨간색 : 틀린 답)

을 알 수 있고, MCAoAN의 경우 이미지와 질의 간의 관계를 적절히 보았기 때문에 전반적으로 개선되었다는 것을 알 수 있다.

5 Analysis

MCAoAN의 경우 현재 학습된 모델의 결과가 기대한 수치보다 낮게 나왔는데, 이는 transformer구조를 end2end로 학

습시키기 위해 64라는 낮은 embedding vector의 차수를 채택했는데, 64 dimension으로부터 생기는 bottleneck problem을 해결하지 못한 것으로 보인다.

또한 MCAoAN의 경우 질의에 방향이 포함된 테스크에 대해 적절히 대답을 못하는 경향성을 보여주었는데, 이는 roi에 대한 embedding vector를 뽑는 과정에서 중복된 이미지가 많이 발생하고, 해당 이미지들에 대한 positional embedding

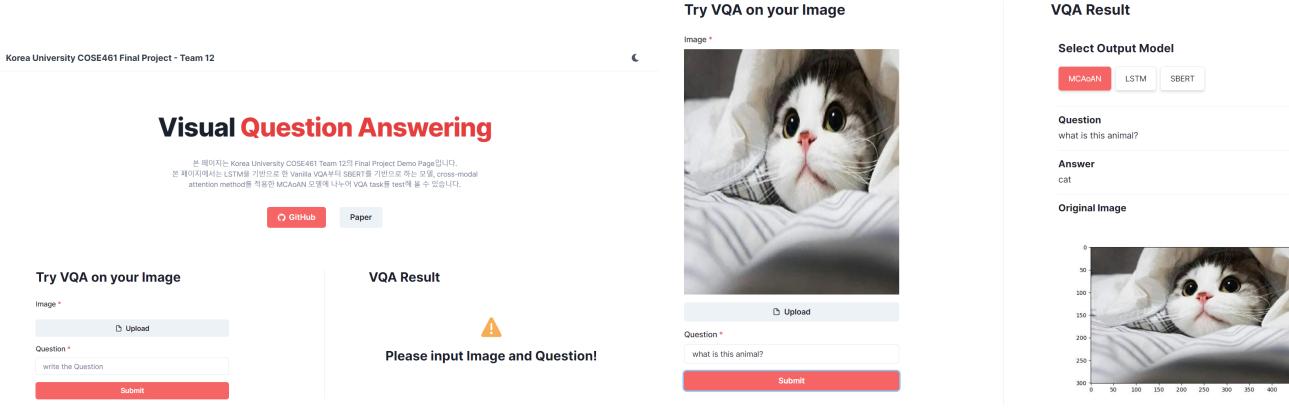
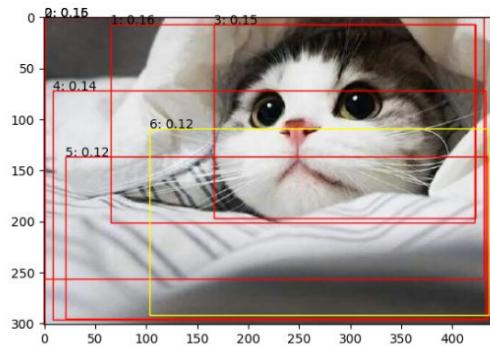


Figure 9: Left : VQA Demo Main Page | Right : VQA Demo Result

Question Word Data

Word	what	is	this	animal	?
Att	0.02	0.03	0.04	0.46	0.45

Image with attention box



Important Attention box list

Attention box list which has attention weight more than avg



Figure 10: Left : VQA Demo Attention | Right : VQA Demo Attention Box List

을 주지 못해서, 해당 이미지의 위치를 찾지 못하는 것으로 보였다.

MCAoAN의 경우 Figure 7에서 볼 수 있듯이 LSTM, SBERT 기반의 모델에 비해 이미지의 내부의 디테일한 질문에 대해 적절히 봐야 하는 곳을 찾고, 정답을 맞추는 모습을 보여주었다.

세 개의 모델 모두가 그림 내의 글자를 묻는 질의에 대해서는 적절히 답변을 못하는 모습을 보였다. 이는 추후에 OCR관련된 모델을 추가해서 적절히 concat하는 방향으로 해결할 수 있을 듯 하다.

6 Conclusion

본 프로젝트를 진행하면서 각 언어 모델이 어떤 상황에서 사용되는지, 그리고 attention 메커니즘과 transformer 구조가 다양한 방향으로 응용할 수 있고, 많은 문제를 해결하기 위해 활용될 수 있다는 것을 배웠다. 특히 이미지 정보의 중요 부분을 attention으로 사용하여 정확도를 끌어올릴 수 있고, cross modal attention을 통해 더 정확한 attention을 구할 수 있으므로 더 큰 성능의 향상이 가능할 것이다.

또한 attention 시각화를 통해 VQA 모델이 어떻게 이미지와 자연어를 이해하고 있는지 알아볼 수 있었다. SBERT기반 모델에서 MCAoAN 모델을 추가적으로 개발했을 때 성능 향상이 기대에 비해 적었지만, explainable한 모델을 개발했다는 것에 의의가 있었다고 판단된다.

시간상의 문제로 Visual Grounding task를 이용한 multi task learning을 진행하지 못한 부분에 대해 아쉬움이 남는다. 추후에 기회가 된다면 multi task learning과 self-supervised learning을 이용해 MCAoAN 모델에 대해 적당한 dimension의 embedding vector을 갖도록 학습시켜볼 계획이다.

7 Demo

본 연구에서 개발한 VQA 모델들을 직접 실행해 볼 수 있는 demo 사이트를 만들었다. demo는 <http://vqateam12.kro.kr/>에서 확인해볼 수 있다.

프론트엔드 개발은 typescript + react기반으로 개발을 진행하였고, 백엔드는 tornado 기반의 kserve(0.8) 프레임 워크를 이용해 개발을 진행하였다.

Figure 9에서 볼 수 있듯이 이미지를 올리고, 영어로 이미지와 관련된 질문을 작성한 후 Submit 버튼을 누르면 각 모델에 대한 추론 결과값이 나오도록 구현하였다.

MCAoAN모델의 경우 Figure 10에서 볼 수 있듯이 질의와 이미지에 대해 attention 시각화를 진행하였고, 이는 모델이 어떤 정보를 중요시하고 있는지 즉각적으로 볼 수 있기 때문에 robust한 데이터로부터 인사이트를 얻는데 도움을 줄 수 있을 것으로 보인다.

References

- [1] Stanislaw Antol Margaret Mitchell C. Lawrence Zitnick Dhruv Batra Devi Parikh Aishwarya Agrawal, Jiasen Lu. Vqa: Visual question answering. 2015.
- [2] Jurgen Schmidhuber Sepp Hochreiter. Long short-term memory. 1997.
- [3] Andrew Zisserman Karen Simonyan. Very deep convolutional networks for large-scale image recognition. 2015.
- [4] Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019.
- [5] Iryna Gurevych Nils Reimers. Sentence-bert: Sentence embeddings using siamese bert-networks. 2019.
- [6] Niki Parmar Jakob Uszkoreit Llion Jones Aidan N. Gomez Łukasz Kaiser Illia Polosukhin Ashish Vaswani, Noam Shazeer. Attention is all you need. 2017.
- [7] Ross Girshick Shaoqing Ren, Kaiming He and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. 2016.
- [8] Chris Buehler Damien Teney Mark Johnson Stephen Gould1 Lei Zhang Peter Anderson, Xiaodong He. Bottom-up and top-down attention for image captioning and visual question answering. 2018.
- [9] Yuhao Cui Dacheng Tao Qi Tian Zhou Yu, Jun Yu. Deep modular co-attention networks for visual question answering. 2019.
- [10] Leonid Sigal Giuseppe Carenini Tanzila Rahman, Shih-Han Chou. An improved attention for visual question answering. 2021.

A Appendix: Team contributions

- 이창수(2016320104) : LSTM, SBERT기반 VQA 모델 개발, 학습 인프라 구성, 모델 학습 관리, 논문 리서치, 데모 백엔드 개발, 코드 형성관리, 전체적인 프로젝트 진행 관리, 보고서 작성
- 강인구 (2016320115) : 논문 리서치, VQA baseline 개발, MCAoAN 모델 개발, 모델 학습 관리, 본 보고서 작성
- 예승형 (2017150358) : 논문 리서치, VQA baseline 개발, 데모 프론트엔드 개발, proposal 발표
- 정경륜(2018320142) : 논문 리서치, VQA baseline 개발, MCAoAN 모델의 Attention visualization 코드 작성, 본 보고서 작성, Proposal 발표 자료 작성, 결과 분석