工學碩士 學位論文

이중 임베딩 기법과 Bi-RNN을 이용한 이미지 캡션에 관한 연구

A Study on Image Caption using Double Embedding Technique and Bi-RNN

指導敎授 徐 東 煥

2018年 8月

韓國海洋大學校 大學院

電氣電子工學科

李 晙 僖

本 論文을 李晙僖 工學碩士 學位論文으로 認准함

委員長 : 工學博士 金 潤 植 ㊞

委 員 : 工學博士 朱 良 翊 ㊞

委 員 : 工學博士 徐 東 煥 ㊞

2018年 8月

韓國海洋大學校 大學院

電氣電子工學科

李 晙 僖

목 차

그림 및 표 목차

<그림목차>

<표 목 차>

이중 임베딩 기법과 Bi-RNN을 이용한 이미지 캡션에 관한 연구

by Jun-Hee, Lee

Department of Electrical & Electronics Engineering
The Graduate School of Korea Maritime and Ocean University
Busan, Republic of Korea

Abstract

본 논문에서는 문장 표현력을 향상시키고 이미지 특징 벡터의 소멸을 방지할 수 있는 이중 Embedding 기법과 문맥에 맞는 문장 순서를 생성하는 Bidirectional Recurrent Neural Network(Bi-RNN)을 적용한 디테일한 이미지 캡션 모델을 제안한다. 이중 Embedding 기법에서, Word Embedding 과정인 Embedding Ⅰ은 캡션의 표현력을 향상시키기 위해 데이터세트의 캡션 단어를 One-hot encoding 방식을 통해 벡터화하고 EmbeddingⅡ는 캡션 생성 과정에서 발생하는 이미지 특징의 소멸을 방지하기 위해 이미지 특징 벡터와 단어 벡터를 융합함으로써 문장 구성 요소의 누락을 방지한다. 또한 디코더 영역은 어휘 및 이미지 특징을 양방향으로 획득하는 Bi-RNN으

로 구성하여 문맥에 맞는 문장의 순서를 학습한다. 마지막으로 인코더와 디코더를 통하여 획득된 전체 이미지, 문장 표현, 문장 순서 특징들을 하나의 벡터공간인 Multimodal 레이어에 융합함으로써 문장의 순서와 표현력을 모두 고려한 디테일한 캡션을 생성한다. 제안하는 모델은 Flickr 8K 및 Flickr 30K, MSCOCO와 같은 이미지 캡션 데이터세트를 이용하여 학습 및 평가를 진행하였으며 객관적인 BLEU와 METEOR 점수를 통해 모델 성능의 우수성을 입증하였다. 그 결과, 제안한 모델은 3개의 다른 캡션 모델들에 비해 BLEU 점수는 최대 20.2점, METEOR 점수는 최대 3.65점이 향상되었다.

A Study on Image Caption using Double Embedding Technique and Bi-RNN

by Jun-Hee, Lee

Department of Electrical & Electronics Engineering
The Graduate School of Korea Maritime and Ocean University
Busan, Republic of Korea

Abstract

This thesis proposes a detailed image caption model that applies the double embedding technique to improve sentence expressiveness and to prevent vanishing of image feature vectors. It uses the bidirectional recurrent neural network (Bi-RNN) to generate a sequence of sentences and fit their contexts. In the double-embedding technique, embedding Ⅰ is a word-embedding process used to vectorize dataset captions through one-hot encoding to improve the expressiveness of the captions. Embedding Ⅱ prevents missed sentence components by fusing image features and word vectors to prevent image features from

vanishing during caption generation. The decoder area, composed of a Bi-RNN that acquires vocabulary and image features in both directions, learns the sequence of sentences that fits their contexts. Finally, through the encoder and decoder, the detailed image caption is generated by considering both sequence and sentence expressiveness by fusing the acquired image features, sentence presentation features, and sentence sequence features into a multimodal layer as a vector space. The proposed model was learned and evaluated using image caption datasets (e.g., Flickr 8K, Flickr 30K, and MSCOCO). The proven BLEU and METEOR scores demonstrate the superiority of the model. The proposed model achieved a BLEU score maximum of 20.2 points and a METEOR score maximum of 3.65 points, which is higher than the scores of other three caption models.

제 1 장 서 론

최근 컴퓨터 비전의 알고리즘들이 적용된 모델들은 이미지 분류 및 이미지 세분화와 같은 다양한 분야에서 뛰어난 성능을 보여주고 있으며 이미지에서 객체를 검출하고 검출된 객체 간의 상호관계를 이해하는 과정을 통해 이미지의 내용을 자연어로 나타내는 이미지 캡션 분야의 연구도 활발히 진행되고 있다.

이미지의 내용을 이해하고 서술하는 작업은 이미지 검색 및 시각 장애인들을 위한 내비게이션에 적용이 가능하며 더 나아가 영상 내에 존재하는 작은 단서에 대해서도 고려가 필요한 영상의학 분야, 범죄 및 CCTV 영상분석과 같은 다양한 분야에 사용될 수 있는 실용적인 가치가 있는 중요한 기술이다. 최근에는 하드웨어 및 소프트웨어의 발달로 인해 이미지에서 정보를 수집하는 단순한 과정들은 기계를 통한 부분적인 자동화[1-4]로 진행되고 있지만 수집된 정보간의 관계를 이해하고 최종적인 판단을 내리는 과정은 아직까지 사람에 의한 수작업으로 이루어진다. 이러한 정보 수집 및 판단에 대한 모든 과정을 자동화하기 위해서는 기계가 이미지에서 객체를 검출하고 검출된 객체 간의 상호관계를 이해하는 과정을 거친 후 이미지의 내용을 스스로 이해하고 판단하는 작업이 필요하다. 이미지의 상세한 내용을 설명하는 작업은 이미지의 각도 및 조명 변화와 같은 외부적인 요소뿐만 아니라 이미지에 존재하는 객체간의 겹침 현상과 자세 변화 같은 내부적인 요소에도 불구하고 사람에게는 비교적 쉬운 일이지만 기계가 사람과 경쟁하기 위해서는 해결해야 하는 많은 과제들이 존재한다[5-10].

이미지 캡션을 위한 초기 연구는 단어 검색 알고리즘을 적용한 이미지 검색 방식을 통해 캡션을 생성하고 이미지에서 추출한 특징 벡터와 해당 이미지의 캡션을 같은 벡터 공간에 위치하도록 학습하여 새로운 이미지에 대한 캡션을 생성하기 위해서는 학습된 이미지와 새로운 이미지 특징의 유사도를 이용하여 가장 유사한 캡션 출력 방식을 사용한다. 이러한 방식은 벡터 공간의 크기가 작은 단순한 키워드를 생성에는 적합하나 벡터 공간의 크기가 큰 문장 생성에는 적합하지 않고 새로운 이미지가 입력될 경우 기존 학습된 이미지와의 유사도를 통해 캡션을 생성하기 때문에 학습된 데이터에 따라 정확도가 크게 달라진다.

최근에는 Convolutional Neural Network(CNN)으로 이루어진 인코더 영역과 Recurrent Neural Network(RNN)으로 구성된 디코더 영역을 가지는 모델[11-15]을 통해 캡션을 생성하는 연구가 진행되고 있다. 아울러 이미지 전체에 대한 특징뿐만 아니라 이미지에서 검출된 객체에 대한 특징을 통해 객체 속성을 추가적으로 이용함으로써 캡션의 표현력을 향상시키는 연구[16]도 진행 중이다. 이러한 연구들은 객체에 대한 수식어를 캡션에 추가함으로써 전체적인 캡션 문장의 길이가 늘어나 표현력이 향상되지만 통해 캡션을 진행하기 때문에 전체적인 캡션 문장은 길어지지만 다수의 객체가 등장하는 이미지에서 객체를 표현하는 속성의 중첩이나 객체의 오검출로 인해 캡션에 수식어가 중복적으로 나열되는 경우가 발생한다.

이미지를 자동으로 검토하기 위한 핵심 과정인 이미지 캡션 모델은 인코더 영역에서 최초로 추출되는 이미지 특징을 이용하여 디코더 영역에서 캡션을 생성하기 때문에 RNN 레이어의 동작 과정에서 발생하는 이미지 특징 벡터 소멸로 인해 해당 특징에 대한 객체가 누락되어 최종 문장에 반영되지 않는 문제가 발생한다. 또한 RNN 레이어는 이전 시간에 생성된 단어를 통해 현재 시간에 나오는 단어를 생성하기 때문에 문장 구성에서

최초로 생성되는 단어가 마지막에 다시 등장할 경우 문장의 길이가 길어질수록 생성된 이전 단어들에 대한 특징이 소멸되어 문장 구조가 파괴되거나 이미지와는 다른 문장이 생성되는 경우가 발생한다.

따라서 본 논문에서는 이중 Embedding 기법을 통해 문장 표현력 향상 및 이미지 특징 벡터의 소멸을 방지하고, Bidirectional Recurrent Neural Network(Bi-RNN)을 적용하여 문맥에 맞는 문장의 순서를 생성함으로써 디테일한 이미지 캡션 모델을 제안한다. 제안한 모델은 이미지 캡션 데이터세트인 Flickr 8K[17] 및 Flickr 30K[18], MSCOCO[19]를 이용하여 학습 및 평가를 진행하였으며 BLEU[20]와 METEOR 점수[21]를 통해 객관적으로 모델의 성능을 나타낸다. 그 결과, 제안하는 이미지 캡션 모델은 비교에 사용된 3개의 다른 캡션 모델에 비해 BLEU 점수는 최대 20.2점, METEOR 점수는 최대 3.65점이 향상되었다.

제 2 장 뉴럴 네트워크 및 평가지표

2.1 Convolutional Neural Network

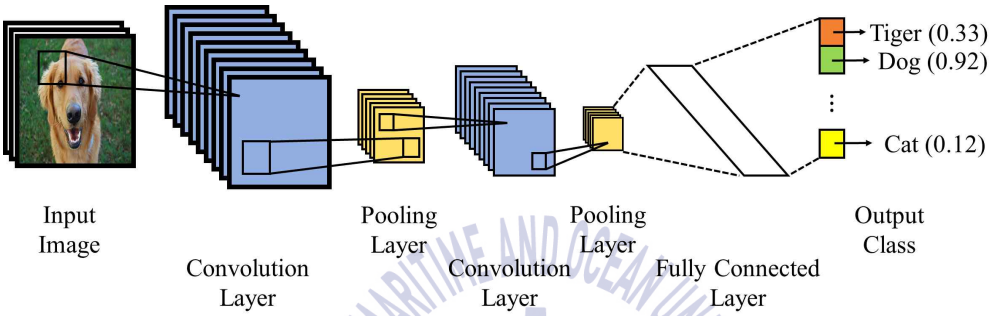


그림 2.1 Convolutional Neural Network 모델 구조

Fig. 2.1 The structure of Convolutional Neural Network Model

일반적으로 Convolutional Neural Network는 이미지를 입력으로 받는 네트워크로 객체 분류 및 검출과 같은 분야에서 우수한 성능을 보이고 있다. 그림 2.1은 기본적인 CNN 모델 구조를 간단하게 나타낸 것이다. CNN의 전처리 과정은 이미지에 존재하는 객체를 사람이 직접 관심영역으로 지정하고 객체의 클래스를 설정하는 작업을 의미하며 다양한 이미지에서 각각의 객체에 대한 관심영역 및 클래스를 지정함으로써 객체들의 각도 및 조명 변화를 학습 할 수 있도록 한다. 학습 단계는 전처리가 완료된 이미지 데이터를 사용하여 CNN의 각 레이어의 연산을 통해 이미지에 대한 특징 맵을 생성한다. CNN은 일반적으로 여러 개의 레이어로 구성되며 대표적으로 Convolution, Pooling, Fully Connected 레이어를 사용한다.

Convolution 레이어는 이미지에 설정해둔 Stride만큼 필터가 이동하면서

이미지와 합성곱 연산을 통해 특징 맵을 생성하는 역할을 담당한다. 이때 이미지 주변에 0을 채워 넣는 Padding과정을 적용함으로써 이미지 가장자리의 정보를 최대한 반영한다. Convolution 레이어는 특징 맵을 생성하기 위한 필터의 종류 및 크기에 따라 생성되는 특징에 영향을 미치며 필터의 크기가 클수록 세밀한 특징을 추출하기가 힘들고 크기가 작을수록 연산시간이 증가하는 단점이 발생한다. 따라서 이미지 데이터의 전체 픽셀크기를 고려하여 필터를 구성하는 것이 필요하다.

Pooling 레이어는 Convolution 레이어의 연산과정을 거쳐 생성되는 특징 맵을 서브 샘플링 과정을 의미한다. 이미지 데이터는 전체 픽셀의 개수가 전체 데이터의 크기와 동일하기 때문에 연산을 통해 생성되는 특징 맵의 크기와 특징의 개수가 매우 많다. 이러한 특징 맵을 그대로 연산에 사용할 경우 모든 특징에 대해 연산을 진행하기 때문에 연산시간이 길어지며 특징 맵의 크기로 인해 네트워크 메모리 용량이 커진다는 단점이 발생한다. Pooling 레이어는 특징 맵에서 일정한 범위에 존재하는 특징들에 대한 최대값(Max), 평균값(Average) 등을 선택하여 중복되거나 비슷한 값을 제거함으로써 특징 맵을 압축하는 역할을 한다. 일반적으로 CNN에서 사용하는 Pooling 레이어는 최대값을 사용하는 Max Pooling 레이어가 많이 활용된다.

Fully Connected 레이어는 일반적인 Artificial Neural Network(ANN)과 같이 동작하며 CNN에서는 다수의 Convolution 레이어와 Pooling 레이어 등을 거치면서 최종적으로 추출된 특징 맵을 신경망 노드의 파라미터를 통해 연결하여 클래스를 분류하기 위해 사용된다. 최초의 파라미터 값들을 통해서는 입력된 특징 맵을 클래스로 정확하게 연결하지 못하지만 관심영역의 객체와 특징 맵을 비교하는 역전파 과정을 통해 학습을 진행하고 파라미터를 갱신하면서 특징 맵과 클래스가 일치하는 확률이 높도록 최적

화된 파라미터를 설정한다.

테스트 단계에서는 새로운 이미지 데이터를 사용하여 학습 단계에서 획득한 각 노드의 최적화된 파라미터를 통해 이미지에 존재하는 객체를 분류하고 높은 확률을 가지는 클래스를 출력하여 모델의 성능을 평가한다. 최근에는 ImageNet에서 우수한 분류 성능을 보인 VGGNet[22], Inception V4[23], ResNet[24]과 같은 대표적인 CNN 모델을 기반으로 객체 검출 분야에서 실시간 객체 검출이 가능한 YOLO[25], RetinaNet[26]과 정밀한 객체 검출이 가능한 Mask R-CNN[27]과 같은 다양한 모델이 존재한다.

머신러닝은 레이어가 깊고 넓을수록 학습 성능이 좋아지지만 기존의 다양한 CNN모델은 Overfitting, Gradient vanishing과 같은 현상으로 인해 레이어를 적층만 해서는 학습 성능이 오히려 저하된다. 본 논문에서 사용하는 Inception V3[28]는 2014년 ImageNet Large Scale Visual Recognition Competition(IRSVRC)에서 좋은 성능을 보인 모델로써 레이어의 적층으로 발생하는 학습 성능 저하를 해결하기 위해 Inception 모듈이라는 새로운 개념의 레이어를 적용하여 레이어 적층으로 인해 발생하는 Overfitting을 방지하고 깊은 모델 구조에서도 역전파 과정을 통한 Weight 갱신이 잘 되도록 함으로써 Gradient vanishing을 해결하여 최고의 성능을 보였다.

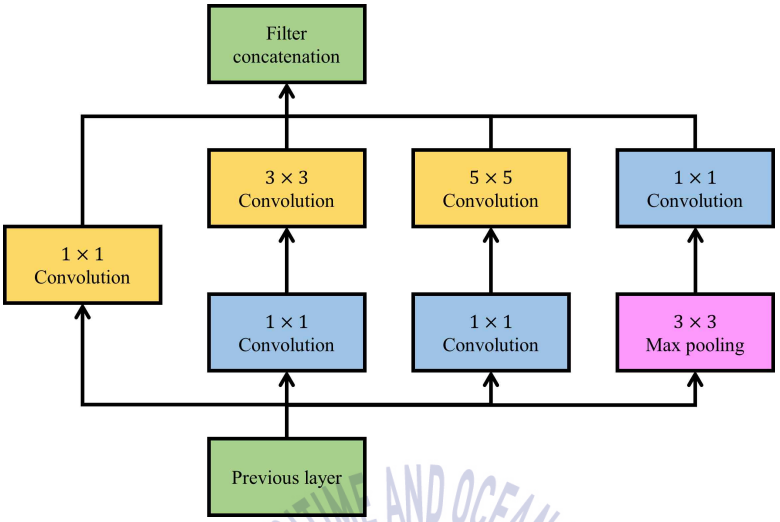


그림 2.2 Inception 모듈 구조

Fig. 2.2 The structure of Inception module

그림 2.2는 Incption 모듈의 구조를 표현한 것이다. Inception 모듈은 Convolution 레이어의 합성곱 연산 결과를 그대로 사용하는 것이 아니라 1×1, 3×3, 5×5 Convolution 레이어를 Convolution 필터처럼 사용하여 특징을 추출한다. 따라서 기존의 Convolution 레이어의 합성곱 연산 이후 Pooling 레이어를 통해 특징 맵의 폭과 높이만 축소하는 방식을 Convolution 필터를 사용하여 특징 맵의 폭과 높이뿐 아니라 필터의 개수도 같이 줄임으로써 차원을 압축한다. 이러한 과정을 통해 각 레이어를 느슨하게 연결함으로써 기존의 Drop Out 레이어의 효과를 레이어 마다 적용함으로써 Overfitting 현상을 해결하고, CNN의 전체적인 연산을 줄여 깊이 있는 학습을 통해 정확도를 향상시켰다.

2.2 Recurrent Neural Network

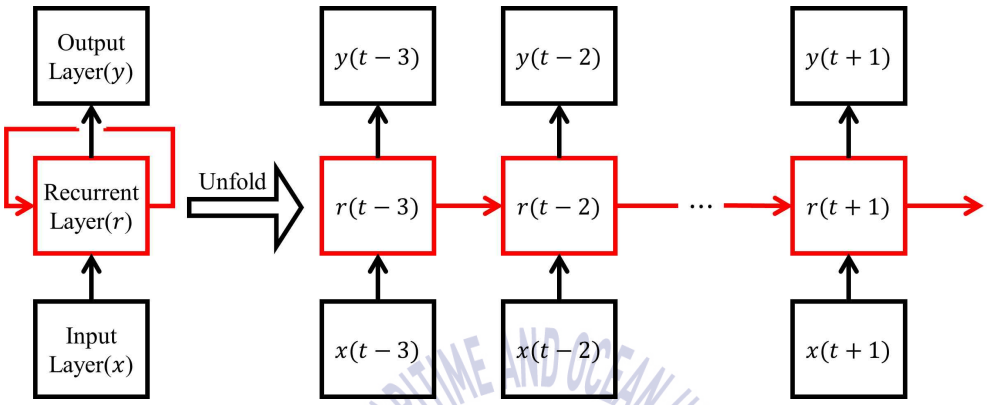


그림 2.3 Recurrent Neural Network 구조

Fig. 2.3 The structure of Recurrent Neural Network

기존의 일반적인 ANN은 모든 입력과 출력이 각각 독립적이라고 가정하며 입력 데이터가 들어오면 출력까지 모든 노드를 한번만 지나가기 때문에 시간에 대한 흐름을 무시하고 학습을 수행하게 된다. 이러한 방식은 시계열 데이터와 같이 순차적인 정보 처리과정이 필요한 데이터를 학습하기에는 적합하지 않다. 그림 2.3은 Recurrent Neural Network(RNN)[29] 구조를 간단하게 나타낸다. RNN은 과거의 출력 값을 Hidden 레이어의 새로운 입력 데이터로 사용하기 때문에 과거의 데이터를 저장하는 메모리를 가지는 것과 같은 방식으로 동작한다. 이러한 방식은 ANN의 구조와 큰 차이점으로 시계열 데이터와 같이 순차적으로 정보를 얻고자 하는 데이터를 입력으로 받는 네트워크로 자연어처리 및 음성인식과 같은 분야에서 우수한 성능을 보이고 있다.

기본적인 RNN은 CNN과 마찬가지로 파라미터 에러의 Gradient를 산출하여 최적의 파라미터를 찾는 Stochastic Gradient Descent(SGD)를 이용하기 때문에 매 스텝마다 Gradient를 산출하여 학습을 진행한다. Gradient 계산 방식은 미분의 Chain rule을 사용하기 때문에 순환 신경망의 Activation function인 Hyperbolic Tangent와 Sigmoid는 양쪽 끝으로 향할수록 Gradient가 0에 수렴하는 Gradient vanishing 현상이 발생하여 파라미터를 갱신하기 위한 오차가 전달되지 못해 긴 시퀀스 데이터에 대해서는 효과적인 학습이 어려워지는 문제가 발생한다. 이러한 학습 문제를 해결하기 위해 CNN에서는 Relu라는 새로운 Activation function을 적용하여 심층학습이 가능하지만 RNN에서는 Relu를 적용하여도 긴 시퀀스 데이터에 대한 장기적인 기억 손실이 발생하는 근본적인 문제 해결이 어렵다는 단점이 있다. 이러한 단점을 해결하기 위해 최근에는 Long Short-Term Memory(LSTM)과 Gated Recurrent Unit(GRU)같은 변형된 RNN이 많이 사용된다.

2.3 Long Short-Term Memory

RNN의 가장 큰 특징은 데이터를 저장하는 메모리를 가짐으로써 시계열 데이터와 같은 순차적인 정보 처리가 가능한 능력이지만 실제 기본적인 RNN은 긴 시퀀스 데이터에 대해서 장기적인 기억 손실 문제가 발생한다. 이러한 문제를 해결하기 위해 제안된 방법이 Long Short-Term Memory(LSTM)[30]이다. LSTM의 기본적인 동작방식은 기존의 RNN과 동일하지만 Cell 내부에 입력, 망각, 출력 게이트를 추가함으로써 장기적인 기억 손실 문제를 해결하였다. 그림 2.4는 LSTM의 Cell 구조를 나타낸다.

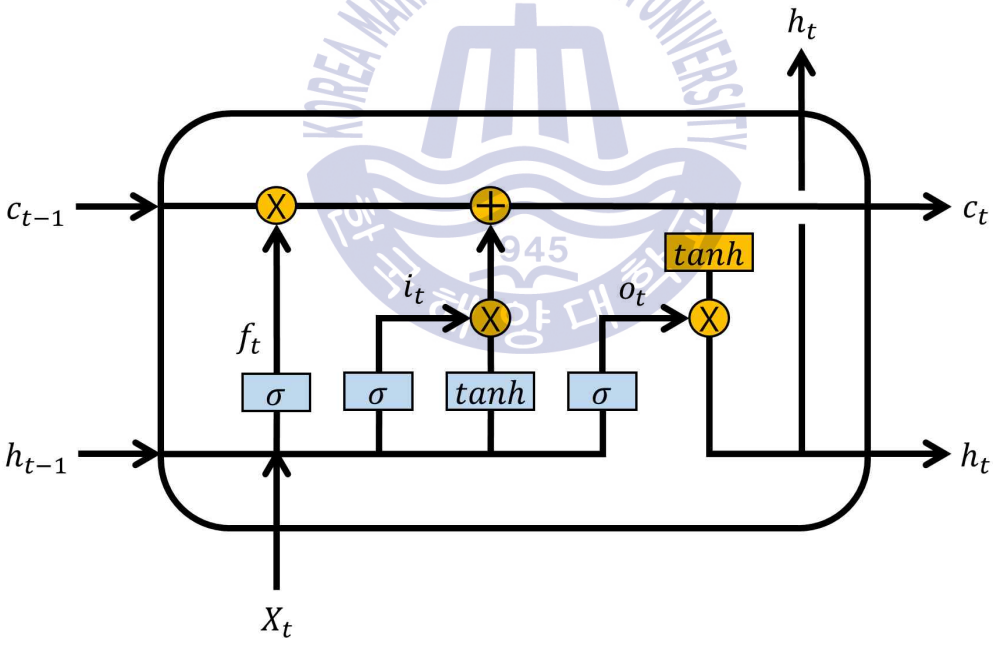


그림 2.4 Long Short-Term Memory Cell 구조

Fig. 2.4 The structure of Long Short-Term Memory Cell

그림에서 c_t는 셀 상태를 의미하며, h_t는 현재 셀의 출력을 i_t, f_t, o_t는

각각 입력, 망각, 출력 게이트를 의미한다. 이를 수식적으로 나타내면 다음과 같다.

$$c_t = f_t \times c_{t-1} + i_t \times \tanh\left(W_c \bullet \left[h_{t-1}, X_t\right] + b_c\right) \tag{2.1}$$

$$h_t = o_t \times \tanh\left(c_t\right) \tag{2.2}$$

$$i_t = \sigma\left(W_i \bullet \left[h_{t-1}, X_t\right] + b_i\right) \tag{2.3}$$

$$f_t = \sigma\left(W_f \bullet \left[h_{t-1}, X_t\right] + b_f\right) \tag{2.4}$$

$$o_t = \sigma\left(W_o \bullet \left[h_{t-1}, X_t\right] + b_o\right) \tag{2.5}$$

입력 게이트는 식 (2.3)을 사용하여 입력 데이터를 조절하고, 망각 게이트는 식 (2.4)를 통해 이전 시간의 셀 상태를 얼마만큼 반영할지를 결정하며 출력 게이트는 식 (2.5)와 같이 출력되는 데이터의 값을 조절한다. 각 식에 등장하는 W는 각 상태에서 적용되는 Weight 값을 의미하며, σ는 Sigmoid를 tanh는 Hyperbolic tangent로써 각 구간에 사용되는 Activation function을 나타낸다. LSTM은 이전 셀 상태에서 망각 게이트를 통해 셀 상태 데이터의 일정량을 소멸하고, 이전 출력 값과 현재의 입력 값을 입력 게이트의 출력과 곱하여 받아들일 입력 데이터를 조절함으로써 식 (2.1)과 같이 현재 셀 상태를 갱신한다. 또한 현재 셀의 출력을 식 (2.2)와 같이 출력 게이트를 곱함으로써 출력 데이터의 양을 조절한다. 이와 같이 LSTM은 이전 셀 상태를 얼마만큼 망각하고 새로운 입력 데이터를 어느

정도 받아들일지 조절함으로써 현재의 셀 상태를 갱신하기 때문에 Gradient vanishing 현상이 없어 긴 시퀀스의 데이터라도 충분한 학습이 가능하다.

2.4 Gated Recurrent Unit

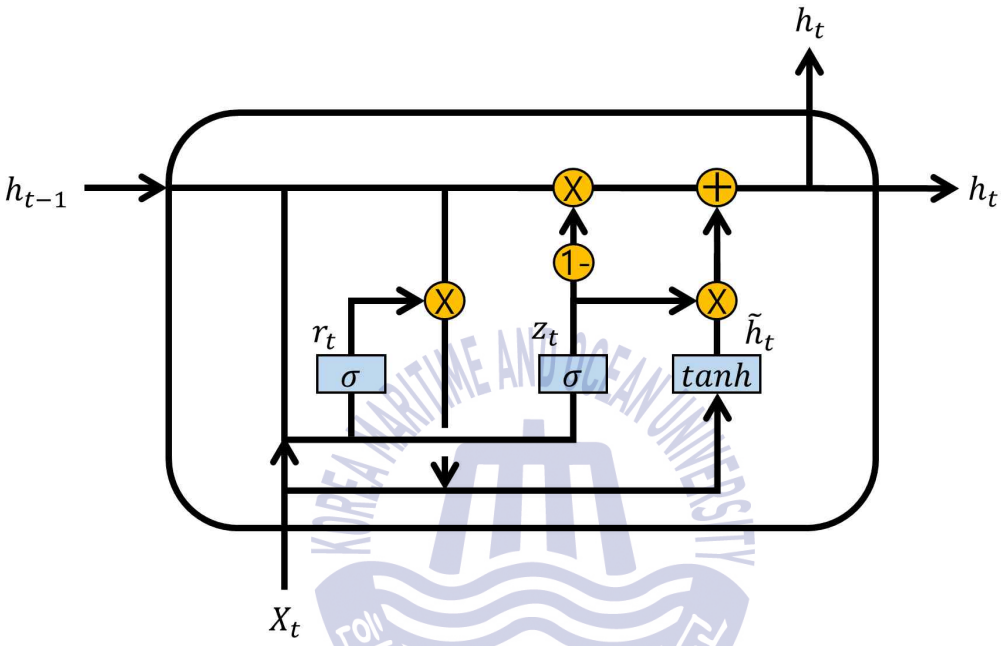


그림 2.5 Gated Recurrent Unit Cell 구조

Fig. 2.5 The structure of Gated Recurrent Unit Cell

Gated Recurrent Unit(GRU)[31]는 RNN의 변형 모델 중 하나로 기존의 LSTM은 입력, 망각, 출력을 담당하는 3개의 게이트를 가지고 있지만 GRU는 입력 및 망각 게이트를 하나의 업데이트 게이트로 통합함으로써 2개의 게이트를 사용하기 때문에 LSTM의 장점을 유지하면서 다양한 게이트로 인해 발생되는 복잡한 과정을 단순화시켰다. 또한 구조의 단순화에도 불구하고 LSTM과 성능이 유사하기 때문에 다양한 분야에서 많이 적용하고 있다. 그림 2.5는 GRU Cell 구조를 나타낸다. 그림에서 r_t는 리

셋 게이트를 z_t는 업데이트 게이트를 나타내며 $\tilde{h}_t$는 현재 Cell이 가지고 있는 데이터를 의미한다. σ와 tanh는 각각 Activation function인 Sigmoid와 Hyperbolic tangent를 나타낸다. GRU Cell의 동작방식을 수식적으로 표현하면 다음과 같다.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{2.6}$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{2.7}$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \times h_{t-1}, x_t]) \tag{2.8}$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t \tag{2.9}$$

각 식에 사용되는 W는 각 구간에서 사용되는 Weight값을 의미한다. 리셋 게이트는 식 (2.6)과 같이 표현되며 현재 시점의 입력 데이터와 이전 시간의 출력 데이터에 대한 식을 Sigmoid함수로 활성화한다. 또한 식 (2.7)은 업데이트 게이트로 Activation function으로 사용하는 Sigmoid의 범위인 0~1사이 값을 사용하여 0일 경우에는 과거 정보를 모두 잊어버리고, 1이면 과거 정보를 모두 사용한다. 이러한 GRU의 2개의 게이트를 이용하여 현재 Cell이 가지는 정보를 식 (2.8)과 같이 도출하며 이때 사용하는 Activation function은 Hyperbolic tangent를 사용한다. 출력 데이터는 식 (2.9)와 같이 현재 Cell의 정보와 과거 Cell의 정보가 합쳐진 데이터가 출력된다.

2.5 Bidirectional Recurrent Neural Network

Bidirectional Recurrent Neural Network(Bi-RNN)[32]은 특정 시점에서의 출력 값이 이전 시점뿐만 아니라 이후 시점의 데이터까지 고려하는 RNN의 확장 모델이다. RNN을 가장 많이 사용하는 자연어처리 연구에서 현재의 단어를 유추할 때 앞서 획득한 정보를 통해 단어를 유추하지만 이후에 나오는 정보도 중요한 역할을 하는 경우가 발생한다. 문장을 예로 들면 “나는 영화관에서 영화를 보았다.” 라는 문장에서 ‘영화’라는 단어를 유추하기 위해서는 앞서 나오는 정보인 ‘영화관’도 중요하지만 이후에 나오는 ‘보았다’라는 표현도 영화를 유추하는데 중요한 정보로 사용이 가능하다. 이처럼 현재 데이터를 처리하기 위해서 양쪽 시점의 데이터를 참고하는 경우에 Bi-RNN을 적용한다. 그림 2.6은 Bi-RNN의 구조를 표현한다.

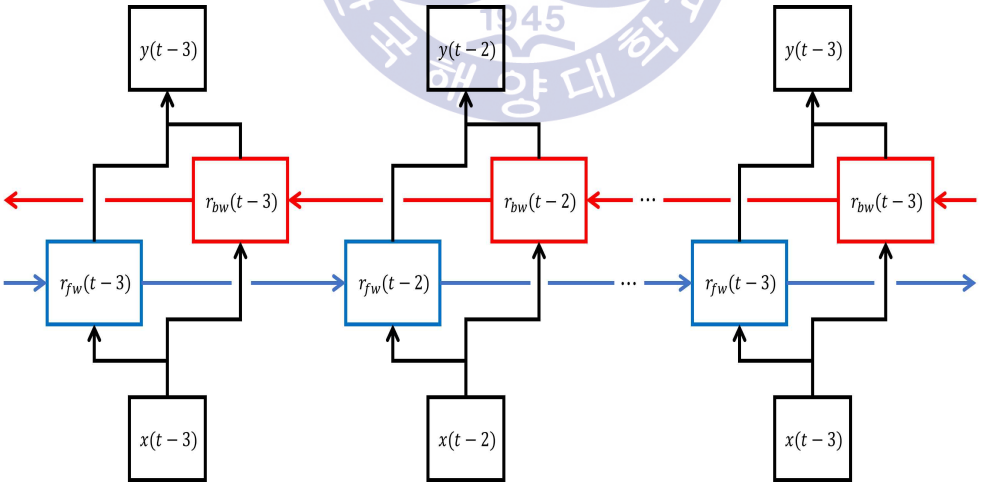


그림 2.6 Bidirectional Recurrent Neural Network 구조

Fig. 2.6 The structure of Bidirectional Recurrent Neural Network

순방향의 출력을 $r_{fw}(t)$, 역방향의 출력을 $r_{bw}(t)$라고 한다면 Bi-RNN의 출력 $y(t)$는 식 (2.10)과 같이 나타낼 수 있다.

$$y(t) = \left[{r_{fw}}^{T}(t) \, , \, {r_{bw}}^{T}(t) \right]^{T} \tag{2.10}$$

Bi-RNN은 시간의 순방향으로 진행하는 RNN과 역방향으로 진행하는 RNN 두 개가 존재하며 두 RNN의 출력을 하나로 합쳐 최종적인 출력으로 사용한다. Bi-RNN을 사용할 경우에는 기존의 RNN에 비해서 한정된 학습 데이터를 사용하여 순방향을 통해 획득하는 이전 시점의 데이터와 역방향을 통해 획득하는 이후 시점의 데이터를 모두 사용하여 학습을 진행한다는 장점을 가지고 있다.

2.6 Bi-Lingual Evaluation Understudy

Bi-Lingual Evaluation Understudy(BLEU)[20]는 자연어 처리 분야에서 기계 번역을 자동으로 평가하기 위해 사용하는 객관적인 성능 평가지표로써 이미지 캡션에서도 많이 사용되고 있다. 이미지 캡션에서 BLEU 평가 방식은 데이터세트에 주어진 캡션 문장을 기준으로 이미지 캡션 모델이 생성하는 문장과 비교를 통해 점수를 산출한다. 다음 식 (2.11)과 식 (2.12)는 BLEU 점수의 산출 방법을 나타낸다.

$$BP = \begin{cases} 1 & \text{if} \quad c > r \\ e^{\left(1-\frac{r}{c}\right)} & \text{if} \quad c \leq r \end{cases} \tag{2.11}$$

$$BLEU = BP \times \exp\left(\frac{1}{N}\sum_{N=1}^{N} \log P_N\right) \tag{2.12}$$

식 (2.11)의 BP는 Brevity Penalty로써 문장의 길이가 짧을 경우 높은 점수를 받는 경우에 대한 패널티를 주기 위해 사용되며, 여기에서 r의 경우에는 데이터세트에서 주어진 캡션의 길이, c는 모델에서 생성하는 캡션의 길이를 의미한다. 식 (2.12)는 최종적인 BLEU 점수를 구하는 식으로써 N은 평가에 사용되는 Gram의 수를 나타내며 P_N은 평가에 사용된 Gram에 대한 정확도를 나타낸다.

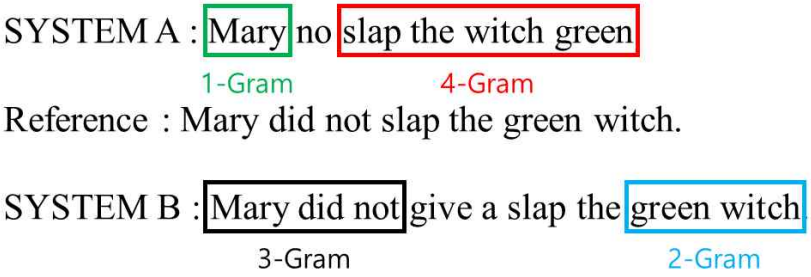


그림 2.7 BLEU 점수 예시 문장

Fig. 2.7 The example sentence of BLEU score

그림 2.7은 BLEU 점수를 산출하기 위한 간단한 예시 문장을 보여준다. BLEU 점수를 산출하기 위해서는 1-Gram ~ 4-Gram을 사용하며 1-Gram은 주어진 문장과 시스템에서 생성하는 문장을 1개의 단어를 이용해 비교를 진행하며 4-Gram은 연속된 4개의 단어를 사용하여 문장 간의 유사도를 비교한다. 표 2.1은 그림 2.7의 예시 문장을 통해 각 Gram에서의 정확도 및 BLEU-4 점수를 산출하였다.

표 2.1 그림 2.7의 BLEU 점수 결과

Table 2.1 The results of Fig. 2.7 BLEU score

	SYSTEM A	SYSTEM B
1-Gram Precision	5/6	7/9
2-Gram Precision	1/5	5/8
3-Gram Precision	0/4	3/7
4-Gram Precision	0/3	1/6
Brevity Penalty	6/7	10/7
BLEU-4	0	0.46

SYSTEM A에 비해 SYSTEM B의 경우 문장의 길이가 길기 때문에 Brevity Penalty인 *BP*가 높으며 Reference 문장과의 4-Gram의 정확도가 SYSTEM A보다 높다. 비교하는 단어의 연속성이 높은 4-Gram이 높을수록 사람이 작성한 Reference 문장과 모델이 생성하는 문장이 비슷하기 때문에 BLEU-4의 점수가 높을수록 문장의 표현력이 사람과 비슷해져 모델의 성능이 좋다고 볼 수 있다.

2.7 Metric for Evaluation of Translation with Explicit ORdering

Metric for Evaluation of Translation with Explicit ORdering(METEOR)[21]는 기계 번역을 평가하는 기준의 하나로써 BLEU에서 발견된 문제점을 수정하였다. METEOR는 BLEU와 마찬가지로 사람이 작성한 문장과 모델이 생성한 문장의 비교를 통해 점수를 산출하지만 BLEU는 코퍼스 수준의 비교를 통해 점수를 산출하는 반면 METEOR는 문장 또는 세그먼트 수준의 평가를 통해 점수를 산출한다. 다음 식은 METEOR 점수 산출 방법을 나타낸다.

$$P = \frac{m}{w_t} \tag{2.13}$$

$$R = \frac{m}{w_r} \tag{2.14}$$

$$F_{mean} = \frac{10PR}{R+9P} \tag{2.15}$$

$$p = 0.5\left(\frac{c}{u_m}\right)^3 \tag{2.16}$$

$$METEOR = F_{mean}(1-p) \tag{2.17}$$

식 (2.13)의 P는 Unigram Precision을 나타내며 m은 생성된 문장에 대해 Reference 문장에서 발견되는 Unigram의 수를 의미하고 w_t는 생성된 문장의 Unigram의 수를 의미한다. 식 (2.14)의 R은 Unigram Recall을 의미하며 m은 식 (2.13)과 동일하고 w_r은 Reference 문장의 Unigram의 수를 나타낸다. Precision과 Recall은 조화평균을 사용하여 식 (2.15)와 같이 결합이 가능하며 Recall의 경우는 Precision에 비해 9배가 높은 가중치를 적용한다. F_{mean}은 한 단어에 대해서만 고려하는 점수이기 때문에 문장 전체를 비교하기 위해서는 BLEU의 Brevity Penalty와 같이 문장의 길이에 따른 페널티가 필요하다. METEOR는 식 (2.16)을 사용하여 n-Gram 매치에 따른 점수를 산출하기 위해 Unigram은 가장 작은 그룹으로 묶이며 여기에서 c는 Reference 문장과 생성된 문장에서 인접한 Unigram의 그룹의 수를 u_m은 매핑 된 Unigram의 수를 의미한다. 식 (2.17)은 최종적인 METEOR 점수를 의미한다. 다음 그림 2.8은 METEOR 점수 산출을 위한 간단한 예시 문장을 표현한다.

Reference : Mary did not slap the green witch.

SYSTEM B : Mary did not give a slap the green witch.

Matching Ref : Mary did not slap the green witch.

Matching SYS : Mary did not slap the green witch.

그림 2.8 METEOR 점수 예시 문장

Fig. 2.8 The example sentence of METEOR score

METEOR 점수는 Reference 문장 및 생성된 문장에서 연속된 Gram간의 매칭을 바로 점수로 사용하는 BLEU와는 다르게 매칭에 따른 문장의 비교를 통해 점수를 산출한다. 표 2.2는 그림 2.8의 예시 문장을 통해 METEOR 점수에 필요한 파라미터를 도출하고 최종 점수를 계산하였다.

표 2.2 그림 2.8의 METEOR 점수 결과

Table 2.2 The results of Fig. 2.8 METEOR score

Parameter	Value
P	7/9
R	7/7
F_{mean}	0.972
$\frac{c}{u_m}$	$\frac{2}{7}$
p	0.012
METEOR	0.96

SYSTEM B의 경우 BLEU에서 사람과의 문장 유사도가 높은 점수인 BLEU-4 점수는 0.46이 나왔지만 METEOR 점수를 사용하여 평가할 경우 0.96이 나오는 것을 확인 할 수 있다. 따라서 METEOR 점수가 높을수록 사람이 작성한 Reference 문장과 유사하다는 것을 확인 할 수 있다.

3. 제안한 이미지 캡션 모델

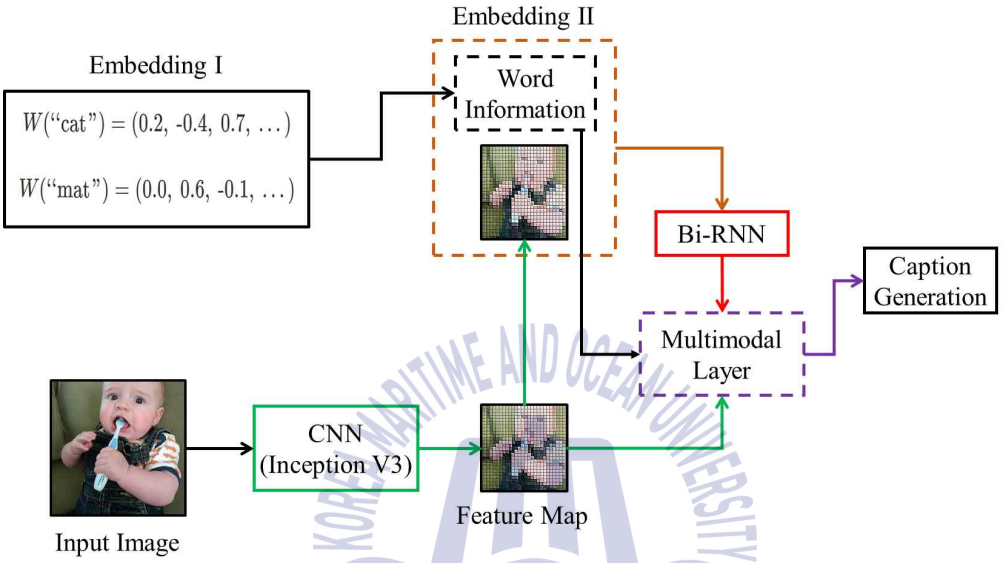


그림 3.1 제안한 이미지 캡션 모델 개념

Fig. 3.1 The concept of proposed image caption model

본 논문에서는 문장 표현력을 향상시키고 이미지 특징 벡터의 소멸을 방지할 수 있는 이중 Embedding 기법과 문맥에 맞는 문장의 순서를 생성할 수 있는 Bidirectional Recurrent Neural Network(Bi-RNN)을 이용하여 디테일한 이미지 캡션 모델을 제안한다. 그림 3.1은 제안한 이미지 캡션 모델의 개념을 보여준다. 인코더 영역인 CNN는 그림 2.1과 같이 CNN의 Fully connected 레이어에서 획득하는 이미지 특징을 디코더 영역에 전송하며 CNN은 ImageNet에서 우수성을 보인 Inception V3를 사용하였다. 이중 Embedding 기법은 Embedding Ⅰ, Embedding Ⅱ로 구성되며 Embedding Ⅰ

은 캡션의 표현력을 향상시키기 위해 데이터세트의 캡션에 주어진 단어를 One-hot encoding을 통해 벡터화하는 Word Embedding 과정을 거친다. EmbeddingⅡ는 Word Embedding 과정을 통해 획득한 단어 정보와 CNN에서 획득하는 이미지 특징을 융합한다. 융합된 데이터는 순방향과 역방향 특징 학습을 통해 문맥에 맞는 문장 순서를 생성하는 Bi-RNN의 입력으로 사용된다. Multimodal 레이어는 CNN의 이미지 특징, Bi-RNN의 문장 순서의 특징, EmbeddingⅡ의 문장 표현력의 특징을 동일한 벡터 공간에 표현하고 Softmax를 통해 이미지 및 문장 순서, 문장 표현이 고려된 디테일한 캡션 생성이 가능하다. 이후 절에서 제안한 이미지 캡션 모델의 각 영역별 구조와 학습에 사용된 데이터세트에 대해 설명한다.

3.1 이중 Embedding 기법과 Bi-RNN을 이용한 캡션 구성 과정

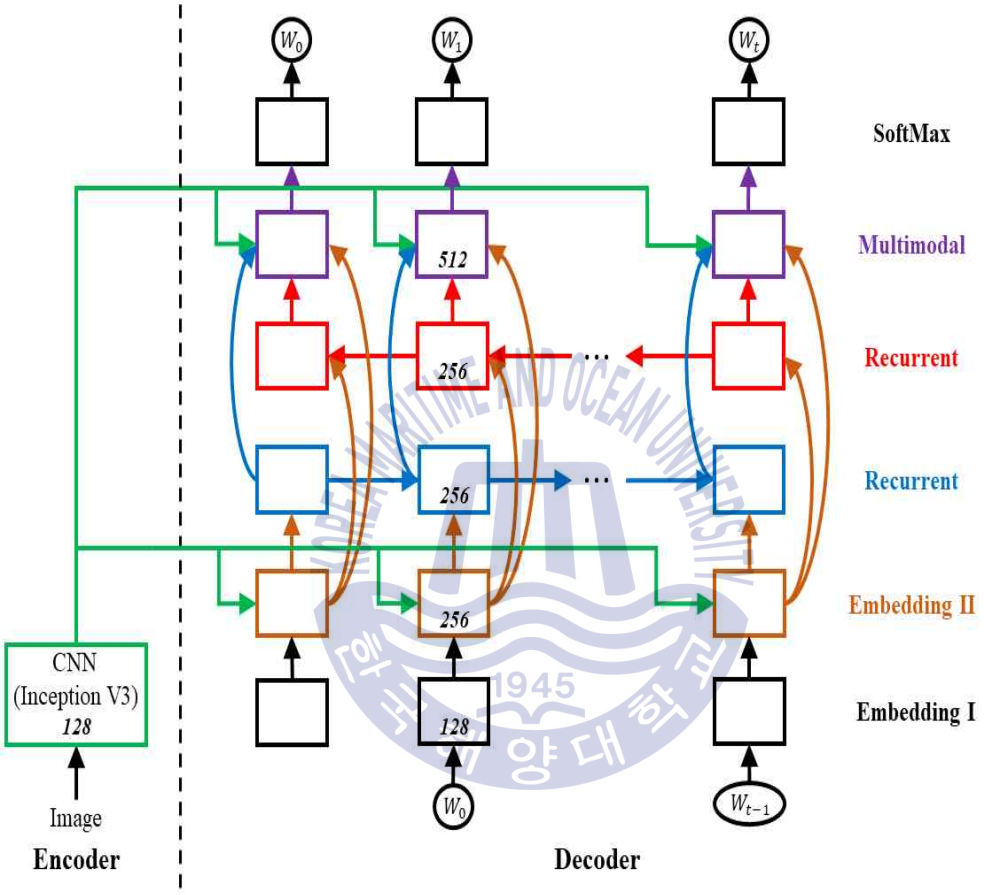


그림 3.2 제안한 이미지 캡션 모델 구조

Fig. 3.2 The structure of proposed image caption model

그림 3.2는 제안한 이미지 캡션 모델의 구조를 나타낸다. 이미지 캡션의 디코더 영역은 인코더 영역의 이미지 특징 맵을 통해 캡션을 생성하는 단계로써 이미지 캡션 모델의 핵심 단계이다. 디코더 영역은 이중

Embedding 기법을 통해 문장의 표현력을 향상시키고 Bi-RNN 학습 과정에서 문맥에 맞는 문장 순서를 구성한다. 캡션 구성 과정은 다음과 같은 세부 단계로 진행된다.

이중 Embedding 기법은 Bi-RNN에 공급되기 전 캡션 표현력을 향상시키는 과정으로 EmbeddingⅠ, EmbeddingⅡ로 구성되며 EmbeddingⅠ은 데이터세트의 캡션에 주어진 단어를 One-hot encoding을 통해 벡터화하는 Word Embedding 과정을 거친다. Word Embedding 단계는 기존의 Word2Vec[33]와 같은 사전 학습된 벡터가 아닌 이미지 캡션 모델의 일부 신경망으로 적용시켜 모델과 학습을 동시에 진행함으로써 이미지를 표현하는 방식에 대한 학습이 가능하여 문장 표현력이 향상된다. 또한 EmbeddingⅡ는 Word Embedding 과정을 통해 획득한 단어 정보와 CNN에서 획득하는 이미지 특징을 융합한다.

Bi-RNN은 이중 Embedding 기법을 통해 생성되는 융합된 벡터를 입력으로 사용하며 본 논문에서는 GRU를 사용하여 Bi-RNN을 구성한다. GRU는 LSTM과 성능의 차이가 미비하지만 내부 게이트의 간소화로 구조적 단순함을 가지기 때문에 내부 연산 속도가 빠르다. 또한 역방향과 순방향의 특징을 통해 현재의 단어를 생성하는 Bidirectional 구조를 적용하여 문맥에 맞는 문장 순서를 통해 단어를 생성한다.

3.2 Multimodal 레이어를 이용한 캡션 생성 과정

Multimodal 레이어는 EmbeddingⅡ 레이어의 출력인 단어 표현을 최종적인 단어 표현에 반영하고, Bi-RNN 모델에서 이미지의 특징과 양방향 특징을 통해 획득한 문장 순서를 각각 2개의 Multimodal 레이어의 입력으로 사용한다. 또한 전체적인 이미지의 특징을 고려하기 위해 CNN의 이미지 특징 벡터를 불러옴으로써 Multimodal 공간에서 각각 3개의 레이어에서 획득하는 데이터를 하나의 동일한 특징 공간에 표현한다. 그림 3.3은 Multimodal 레이어의 구조를 표현한다.

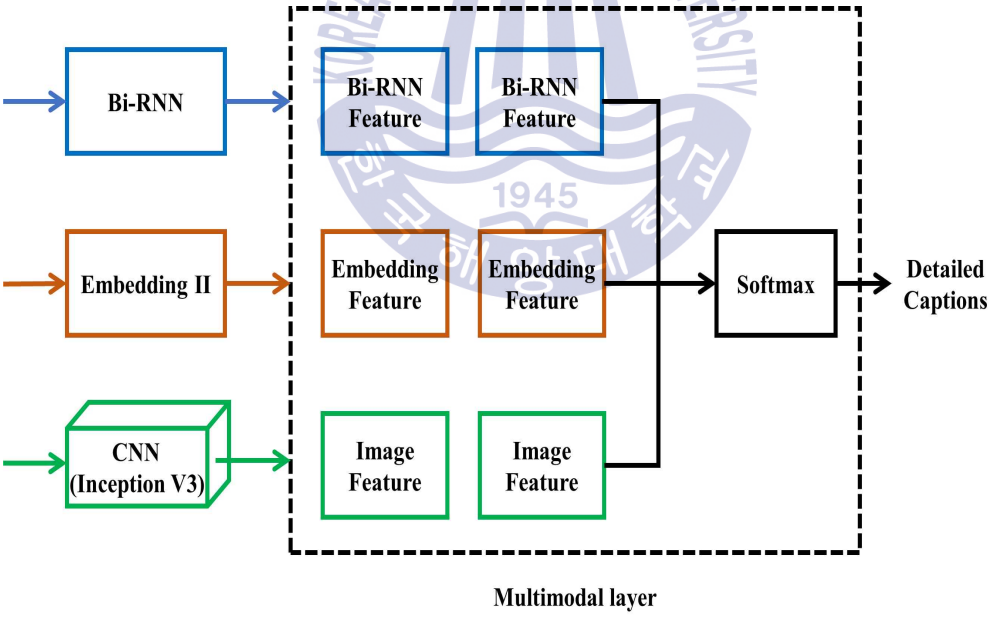


그림 3.3 Multimodal 레이어 구조

Fig. 3.3 The structure of Multimodal layer

Multimodal 레이어는 식 (3.1)과 같이 정의된다.

$$m(t) = g(V_\omega \cdot \omega(t) + V_{br} \cdot br(t) + V_I \cdot I(t)) \quad (3.1)$$

$m(t)$는 Multimodal 레이어를 의미하며 V_w , V_{br} , V_I는 각각 Word embedding, Bi-RNN, 이미지 특징 벡터를 뜻한다. g는 다음 단어의 확률 분포를 예측하는 Softmax를 나타낸다.

제 4 장 실험 및 결과

4.1 데이터세트 및 전처리 과정

본 절에서는 제안한 이미지 캡션 모델의 학습 및 검증을 위해 Flickr 8K와 Flickr 30K, MS COCO와 같은 이미지 캡션 분야에서 많이 사용되는 데이터세트에 대해 설명한다. 이 데이터세트는 학습 및 검증, 테스트 세트로 나누어지며 다음 표 4.1과 같이 테스트 세트를 분류하여 적용한다.

표 4.1 벤치마크 데이터세트의 분류

Table 4.1 The categorizing of the benchmark datasets

Dataset	Training set	Validation set	Test set
Flickr 8K	6,000	1,000	1,000
Flickr 30K	28,000	1,000	1,000
MSCOCO	82,783	40,504	40,775

Flickr 8K는 각 이미지에 5개의 캡션 문장이 주석으로 제공되며 85:5:10 비율로 학습, 검증, 테스트용 데이터세트를 분류한다. 또한 학습 데이터세트에서 전처리 과정으로 단어 필터링 과정을 거치며 전체 캡션 문장에서 5회 미만으로 등장하는 단어는 Unkown 데이터로 분류하여 캡션 생성에서 해당 단어를 사용하는 것을 배제하도록 필터링한다. Flickr 30K

는 Flickr 8K의 확장형으로 구조는 Flickr 8K와 동일하다. 또한 MSCOCO의 경우에도 각 이미지에 5개의 문장이 주어지며 데이터 전처리 과정은 Flickr 8K와 동일하게 처리한다.

표 4.2 어휘가 다른 단어 유형의 수

Table 4.2 The number of different word types in the vocabulary

Dataset	Number of different word type
Flickr 8K	2,539
Flickr 30K	7,415
MSCOCO	8,792

표 4.2는 전처리 과정을 통해 각 데이터세트에서 사용된 어휘가 다른 단어 유형의 수를 나타낸다. Flickr 8K의 경우에는 2,539개의 단어를 Flickr 30K는 7,415개, MSCOCO는 8,792개의 단어를 가지고 있다.

4.2 실험 결과 분석

제안하는 모델이 캡션 생성 과정에서 발생하는 이미지 벡터 크기 감소에 얼마만큼 강인한지 여러 모델들이 사용하는 방식과의 비교를 그림 4.1과 같이 나타낸다.

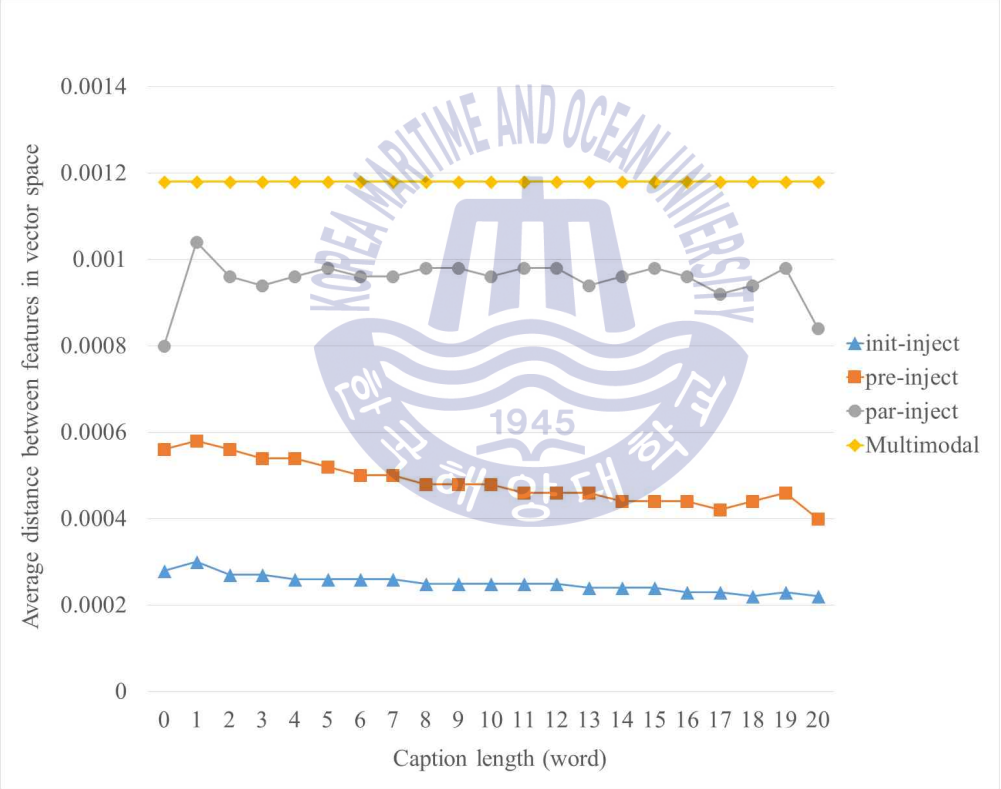


그림 4.1 캡션 생성으로 인한 이미지 벡터 크기 감소

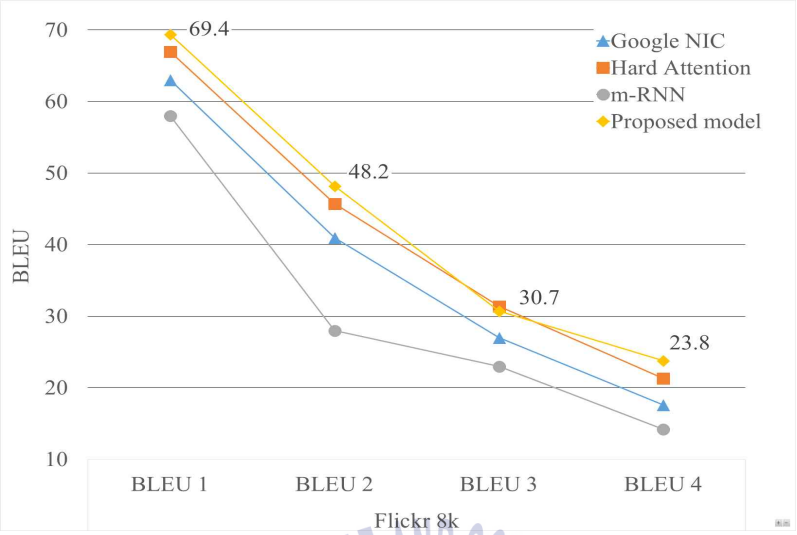
Fig. 4.1 Decrease of image vector size due to caption generation

Initial inject 방식은 초기 캡션 생성 모델에서 사용하는 방식으로 이미지 특징 벡터를 RNN의 초기 상태 벡터로 사용되며 RNN을 초기화 한 후 단어 벡터를 통해 캡션을 생성한다. 초기 상태 벡터의 크기에 따라 이미지 벡터의 크기가 결정되기 때문에 이미지 벡터의 평균 거리가 제일 낮아 가장 작은 이미지 벡터 크기를 가지는 것을 보여준다.

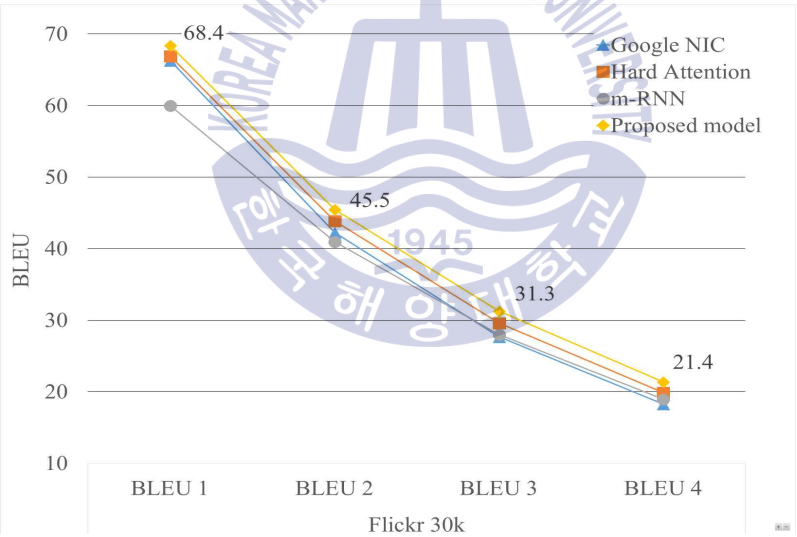
또한 Prefix inject 방식은 RNN의 첫 번째 입력으로 이미지를 사용하는 방식이며 첫 번째 이미지가 입력으로 사용 된 후 두 번째부터 단어 벡터를 사용하여 캡션을 생성한다. 이 방식은 Initial inject 방식에 비해 단어 벡터와 동일한 크기의 이미지 벡터를 사용하기 때문에 이미지 벡터들의 거리가 좀 더 멀어지고 따라서 전체 이미지 벡터의 크기가 커지는 것을 알 수 있다. 하지만 이 두 방식의 경우 레이어의 순환에 따라 캡션 생성이 진행될수록 지속적인 이미지 벡터의 크기 감소를 확인할 수 있다.

본 모델에서 사용하는 Parallel inject 방식은 기존의 이미지 특징 벡터를 단어 벡터와 병렬로 RNN의 입력에 사용하는 2개의 개별 입력이 아닌 이중 Embedding 기법을 통해 이미지 벡터와 단어 벡터를 결합하여 단일 입력으로 사용한다. 이때 입력되는 이미지 벡터는 동일한 이미지 벡터가 사용된다. 단어 벡터가 입력으로 사용될 때 이미지 벡터도 함께 입력되기 때문에 캡션 생성 과정에서 이미지 벡터가 계속 공급되어 벡터의 크기가 감소하다 증가하는 모습을 보이며 캡션의 길이가 늘어나도 전체적인 벡터의 크기 감소가 다른 방식에 비해 일정한 것을 알 수 있다.

Multimodal 레이어는 RNN의 학습 이후에 적용되기 때문에 RNN의 동작특성으로 인해 발생하는 이미지 벡터의 크기 감소가 나타나지 않아 제일 많은 이미지 특징 벡터 크기를 유지하는 것을 알 수 있다. 본 모델에서는 Multimodal 레이어의 이미지 특징 벡터를 최종 캡션 생성에서 사용함으로써 이미지 전체 특징을 고려한 캡션을 생성 할 수 있도록 하였다.



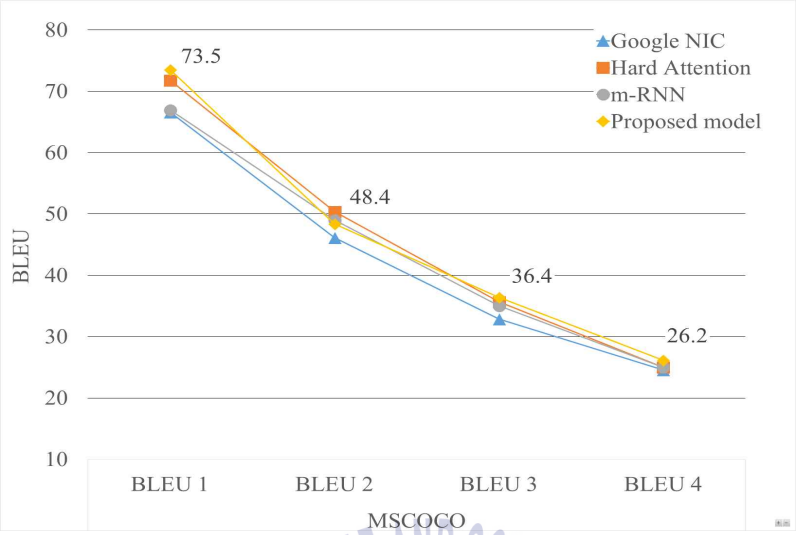
(a)



(b)

그림 4.2 데이터세트의 모델별 BLEU 점수 결과 :
(a) Flickr 8K, (b) Flickr 30K, (c) MSCOCO

Fig. 4.2 The results of BLEU score by models on the dataset :
(a) Flickr 8K, (b) Flickr 30K, (c) MSCOCO



(c)

그림 4.2 데이터세트의 모델별 BLEU 점수 결과(계속) : (a) Flickr 8K, (b) Flickr 30K, (c) MSCOCO

Fig. 4.2 The results of BLEU score by models on the dataset(cont.) : (a) Flickr 8K, (b) Flickr 30K, (c) MSCOCO

그림 4.2는 각 데이터세트의 모델별 BLEU 점수를 나타낸다. 제안하는 모델의 성능을 기존의 3개의 모델과 비교하였으며 디코더 영역을 GRU로 구성하고 여러 종류(Flickr 8K, Flickr 30K, MSCOCO)의 데이터세트를 통해 학습을 진행하여 제안하는 모델이 기존 캡션 모델에 비해 우수한 성능을 보인다.

표 4.3 데이터세트의 모델별 BLEU 점수 결과

Table 4.3 The results of BLEU score by models on the dataset

Dataset	Model	BLEU			
		B1	B2	B3	B4
Flickr 8K	Google NIC[11]	63.0	41.0	27.0	17.6
	Hard Attention[12]	67.0	45.7	**31.4**	21.3
	m-RNN[13]	58.0	28.0	23.0	14.2
	Proposed model	**69.4**	**48.2**	30.7	**23.8**
Flickr 30K	Google NIC[11]	66.3	42.3	27.7	18.3
	Hard Attention[12]	66.9	43.9	29.6	19.9
	m-RNN[13]	60.0	41.0	28.0	19.0
	Proposed model	**68.4**	**45.5**	**31.3**	**21.4**
MSCOCO	Google NIC[11]	66.6	46.1	32.9	24.6
	Hard Attention[12]	71.8	**50.4**	35.7	25.0
	m-RNN[13]	67.0	49.0	35.0	25.0
	Proposed model	**73.5**	48.4	**36.4**	**26.2**

표 4.3은 각 모델별 BLEU 점수 결과를 세밀하게 확인하기 위해 점수를 모두 표기하였다. Flickr 8K 경우 BLEU-3에서 Hard Attention 모델에 비해 점수가 낮지만 BLEU-4에서 다른 모델들에 비해 높은 점수를 가져 캡션 성능이 우수한 것을 객관적으로 확인 가능하다. 또한 Flickr 30K의 경우 전체적으로 제안하는 모델의 성능이 우수하며 MSCOCO의 경우에는 Flickr 8K와 비슷하게 BLEU-2에서 Hard Attention 모델에 비해 점수가 낮지만 4-Gram을 이용한 평가 방식인 BLEU-4에서 점수가 높게 나옴으로써 캡션 문장의 표현력이 사람의 표현과 유사하게 표현되는 것을 알 수 있다.

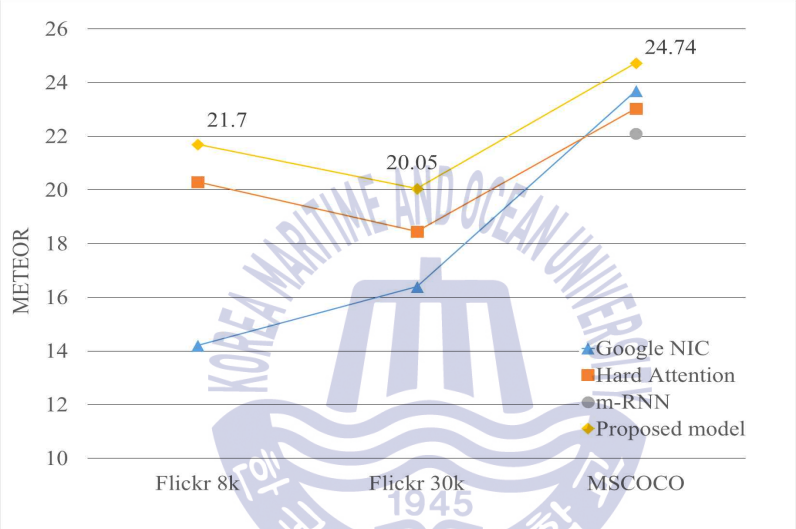


그림 4.3 데이터세트의 모델별 METEOR 점수 결과

Fig. 4.3 The results of METEOR score by models on the dataset

표 4.4 데이터세트의 모델별 METEOR 점수 결과

Table 4.4 The results of METEOR score by models on the dataset

Dataset	Model	METEOR
Flickr 8K	Google NIC[11]	14.20
	Hard Attention[12]	20.30
	m-RNN[13]	-
	Proposed model	**21.70**
Flickr 30K	Google NIC[11]	16.40
	Hard Attention[12]	18.46
	m-RNN[13]	-
	Proposed model	**20.05**
MSCOCO	Google NIC[11]	23.70
	Hard Attention[12]	23.04
	m-RNN[13]	22.10
	Proposed model	**24.74**

그림 4.3과 표 4.4는 각 모델별 METEOR 점수 결과를 나타낸다. BLEU의 문제점을 개선시킨 METEOR 점수의 경우 제안하는 모델이 기존의 다른 모델들에 비해 모든 데이터세트에서 우수한 성능을 가지는 것을 확인할 수 있다. BLEU와 METEOR 점수의 결과를 통해 제안하는 모델은 캡션 생성 과정에서 발생하는 이미지 특징 벡터의 소멸을 이중 Embedding 기법을 통해 방지함으로써 이미지 자체가 가지는 정보를 기존의 모델에 비해 풍부하게 획득 할 수 있으며, 일반적인 RNN이 아닌 Bi-RNN으로 디코더 영역을 구성함으로써 이전 단어 및 이후 단어의 영향을 고려하여 캡션 생성 과정에서 전체 문맥에 맞춰 현재의 단어들을 수용하는 것을 알 수 있다. 그림 4.4는 모델별 생성되는 자막의 샘플을 보여준다.

(a) (b)

(c) (d)

(a)
Reference : A boy in blue jumps his skateboard off some steps while his friends watch.
Google NIC : a man riding a skateboard over a hurdle.
Hard attention : a man riding a skateboard down a snow covered slope.
m-RNN : a man is sitting on a skateboard and people are sitting.
Proposed model : a man in blue is riding a skateboard and people are sitting on bench.

그림 4.4 모델에 의해 생성되는 자막의 샘플

Fig. 4.4 The samples of captions generated by models

(b)

Reference : A blond woman in a blue shirt appears to wait for a ride.
Google NIC : a young boy wearing a red shirt is playing with a basket ball.
Hard attention : a person holding a cell phone in their hand.
m-RNN : a woman is standing outside holding a cell phone.
Proposed model : a woman in a blue shirt is holding a cell phone and is in the driveway.

(c)

Reference : A group of people ride bikes while holding onto large trash bags.
Google NIC : a man wearing a helmet rides a bike.
Hard attention : a group of people riding bikes down a street.
m-RNN : a group of people ride bikes on the road.
Proposed model : a group of people riding bike have trash bags on the road.

(d)

Reference : Four young kids have a picture taken of them while in midair.
Google NIC : a group of people are playing soccer.
Hard attention : a couple of young men playing a game of frisbee.
m-RNN : a group of people around playing outside.
Proposed model : a group of four children jumping in midair.

그림 4.4 모델에 의해 생성되는 자막의 샘플(계속)

Fig. 4.4 The samples of captions generated by models(Cont.)

그림 (a)는 파란색 옷을 입은 소년이 스케이트보드를 타고 있으며, 옆에서 친구 두 명이 그 모습을 지켜보고 있는 이미지이다. Google NIC 모델의 경우 스케이트를 타는 남자 앞에 있는 난간을 허들로 잘못 인식하여 남자가 스케이트보드를 타고 허들을 넘는다는 잘못된 표현을 하며, 옆에 있는 두 명의 사람은 문장에 등장하지 않는다. Hard attention 모델의 경우에는 눈이 덮인 슬로프에서 스케이트보드를 타고 있다고 잘못된 표현을 생성한다. 전체적인 모델들이 보드를 타는 사람에 대해서는 표현이 잘 진행되고 있으나, 옆에 있는 사람에 대해서는 캡션 생성과정에서 이미지 특징 벡터의 소멸로 인해 캡션에 등장하지 않는다. 제안한 모델은 이미지

특징 벡터의 소멸에 강인한 모델의 특성으로 인하여 보드를 타고 있는 사람의 옷 색과 벤치에 앉아있는 사람을 정확하게 표현하였다.

그림 (b)는 주어진 Reference 문장이 파란 셔츠를 입은 금발의 여자가 탈것을 기다리고 있다는 내용의 이미지로 다른 3개의 모델에서는 여자 옷에 대한 색 표현이 캡션에 등장하지 않고, 현재 이미지의 환경에 대한 표현도 등장하지 않는다. 제안한 모델은 여성의 옷의 색과 현재 이미지의 환경이 도로에서 발생하는 것까지 정확하게 표현함으로써 Multimodal Layer를 이용한 전체 이미지 특징을 사용하는 것을 확인 할 수 있다.

그림 (c)에서는 Google NIC와 같은 예전 모델의 경우 단일객체에 대한 표현에 이미지 특징을 모두 소모하여 제일 처음에 나오는 자전거를 타는 사람의 헬멧과 같은 세부적인 특징은 캡션에 등장하지만 다른 사람들은 표현되지 않는다. 제안하는 모델은 자전거를 타는 사람과 쓰레기봉투 및 환경에 대한 표현까지 세밀한 캡션 생성이 가능한 것을 확인 할 수 있다.

마지막으로 그림 (d)는 4명의 어린이가 공중에 떠 있는 이미지로 Google NIC의 경우는 풀을 통해 축구를 하는 상황이라고 유추를 하여 캡션 표현이 이미지의 내용과는 전혀 다르게 출력되며, Hard attention의 경우 학습데이터의 원반던지기와 관련된 이미지에서 사람 또는 강아지가 공중에 떠 있는 장면이 많아 원반던지기를 하는 상황이라는 캡션을 출력한다. 제안하는 모델은 정확하게 4명의 어린이가 공중에 점프하고 있는 상황을 캡션으로 출력함으로써 모델의 캡션 표현이 전체 이미지의 특징 및 특징 소멸에 강인함을 보여준다.

제 5 장 결 론

본 논문에서는 문장 표현력을 향상시키고 이미지 특징 벡터의 소멸을 방지할 수 있는 이중 Embedding 기법과 문맥에 맞는 문장의 순서를 생성하는 Bi-RNN을 적용한 디테일한 이미지 캡션 모델을 제안하였다. 이 모델은 이중 Embedding 기법을 통해 Embedding Ⅰ은 캡션의 표현력을 향상시키기 위해 데이터세트의 캡션을 One-hot encoding방식을 통해 단어를 벡터화하는 Word Embedding 과정을 거치며 EmbeddingⅡ는 캡션 생성 과정에서 발생하는 이미지 특징 벡터의 소멸을 방지하기 위해 이미지 특징 벡터를 단어 벡터와 융합함으로써 문장 구성 요소 누락을 방지한다. 또한 양방향에서 획득하는 어휘 및 이미지 특징을 이용하는 Bi-RNN으로 디코더 영역을 구성하여 문맥에 맞는 문장의 순서를 학습한다. 마지막으로 Multimodal 레이어에 전체 이미지 특징, 이중 Embedding 기법으로 획득하는 문장 표현의 특징, Bi-RNN 학습 과정에서 구성되는 문장 순서의 특징을 하나의 벡터 공간에 표현함으로써 이미지 및 문장의 순서, 문장의 표현력을 모두 고려한 디테일한 캡션을 생성한다. 제안하는 모델은 BLEU와 METEOR 점수를 통해 모델의 성능을 객관적으로 비교하였고 3개의 다른 기존의 캡션 모델에 비해 BLEU 점수는 최대 20.2점, METEOR 점수는 최대 3.65점이 향상되어 제안한 모델의 우수함을 보였다.

이 연구를 통하여 향후 영상의학 및 법의학 분야의 수준 높은 데이터세트가 구축되고 이를 학습한다면 해당 분야에 필요한 캡션을 생성함으로써 이미지의 자동 주석이나, 사용자의 질문에 대한 간단한 답변 표현에 적용이 가능 할 것으로 기대된다.

참 고 문 헌

[1] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, “Convolutional neural networks for large-scale remote-sensing image classification,” IEEE Trasactions on Geoscience and Remote Sensing, vol. 55, no. 2, pp.645-657, 2017.

[2] L. D. Nguyen, D. Lin, Z. Lin, and J. Cao, “Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation,” IEEE International Symposium on Circuits and Systems(ISCAS), pp. 1-5, 2018.

[3] S. Yu, S. Jia, and C. Xu, “Convolutional neural networks for hyperspectral image classification,” Neurocomputing, vol. 219, pp. 88-98, 2017.

[4] P. Morales-Alvarez, A. Perez-Suay, R. Molina, and G. Camps-Valls, “Remote sensing image classification with large-scale gaussian processes,” IEEE Transactions on Geoscience and Remote Sensing, vol. 56, no. 2, pp. 1103-1114, 2018.

[5] B. Gecer, G. Azzopardi, and N. Petkov, “Color-blob-based COSFIRE filters for object recognition,” Image and Vision Computing, vol. 57, pp. 165-174, 2017.

[6] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 4, pp. 834-848, 2018.

[7] K. Grm, V. Stuc, A. Artiges, M. Caron, and H. K. Ekenel, “Strengths and weaknesses of deep learning models for face recognition against image degradations,” IET Biometrics, vol. 7, no. 1, pp. 81-89, 2017.

[8] J. Cleveland, D. Thakur, P. Dames, C. Phillips, T. Kientz, K. Daniilidis, and V. Kumar, “Automated system for semantic object labeling with soft-object recognition and dynamic programming segmentation,” IEEE

Transactions on Automation Science and Engineering, vol. 14, no. 2, pp. 820-833, 2017.

[9] X. Yang, W. Wu, K. Liu, P. W. Kim, A. K. Sangaiah, and G. Jeon, "Long-distance object recognition with image super resolution: A comparative study," IEEE Access, vol. 6, pp. 13429-13438, 2018.

[10] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, "Classification with an edge: improving semantic image segmentation with boundary detection," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 135, pp. 158-172, 2018.

[11] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," IEEE conference on Computer Vision and Pattern Recognition(CVPR), pp. 3156-3164, 2015.

[12] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," International Conference on Machine Learning, pp. 2048-2057, 2015.

[13] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)," arXiv preprint arXiv:1412.6632, 2014.

[14] X. Chen and C. Lawrence Zitnick, "Mind's eye: A recurrent visual representation for image caption generation," IEEE conference on Computer Vision and Pattern Recognition(CVPR), pp. 2422-2431, 2015.

[15] C. Liu, F. Sun, C. Wang, F. Wang, and A. Yuille, "MAT: A multimodal attentive translator for image captioning," arXiv preprint arXiv:1702.05658, 2017

[16] P. Kinghorn, L. Zhang, and L. Shao, "A region-based image caption generator with refined descriptions," Neurocomputing, vol. 272, pp. 416-424, 2018.

[17] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," Journal of Artificial Intelligence Research, vol. 47, pp. 853-899, 2013.

[18] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions,” Transactions of the Association for Computational Linguistics, vol. 2, pp. 67-78, 2014.

[19] T. Y. Lin, M. Maire, S. Belongje, J. Hays, P. Perona, D. Ramanan, and C. L. Zitnick, “Microsoft coco: Common objects in context,” arXiv preprint arXiv:1405.0312, 2014.

[20] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” Proceedings of the 40th annual meeting on asscociation for computational linguistics, pp. 311-318, 2002.

[21] S. Banerjee and A. Lavie, “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments,” Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pp. 65-72, 2005.

[22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXive:1409.1556, 2014.

[23] C. Szegedy, S. loffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” arXiv preprint arXiv:1602.07261, 2016.

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

[25] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” IEEE conference on Computer Vision and Pattern Recognition(CVPR), pp. 7263-7271, 2017.

[26] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” arXiv preprint arXiv:1708.02002, 2017.

[27] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask r-cnn,” IEEE International conference on Computer Vision(ICCV), pp. 2980-2988, 2017.

[28] C. Szegedy, V. Vanhoucke, S. loffe, J. Shlens, and Z. Wojna,

"Rethinking the inception architecture for computer vision," IEEE conference on Computer Vision and Pattern Recognition(CVPR), pp. 2818-2826, 2016.

[29] T. Mikolov, M. Karaflat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," Eleventh Annual Conference of the International Speech Communication Association, 2010.

[30] S. Hochreiter and J. Schnidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[31] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555, 2014.

[32] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, 1997.

[33] Y. Goldberg and O. Levy, "word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method," arXiv preprint arXiv:1402.3722, 2014.