Journal of Korea Game Society 2018 Oct; 18(5): 83-92
http://dx.doi.org/10.7583/JKGS.2018.18.5.83

게임 콘텐츠, 산업, 마케팅

딥 러닝을 이용한 자동 댓글 생성에 관한 연구

최재용, 성소윤, 김경철
한국산업기술대학교 게임공학과
{gclock93, tjdthdbs12, ken}@kpu.ac.kr

A Study on Automatic Comment Generation Using Deep Learning

Jae-yong Choi, So-yun Sung, Kyoung-chul Kim
Dept. of Game & Multimedia Engineering, Korea Polytechnic University

요 약

최근 다수의 분야에서 딥 러닝을 통한 연구 성과들이 사람의 판단력에 근접하는 결과를 보여주고 있다. 그리고 게임 산업에서는 온라인 커뮤니티, SNS의 활성화가 게임 흥행 여부를 결정할 정도로 중요성이 높아지고 있다. 본 연구는 딥 러닝을 이용해 온라인 커뮤니티, SNS에서 활동할 수 있는 시스템을 구성하고, 온라인 공간에서 사람들이 작성한 텍스트를 읽고 그에 대한 반응을 생성하고 스케쥴에 따라 트위터에 올리는 것을 목표로 한다. 순환 신경망(Recurrent Neural Network)을 이용해 텍스트를 생성하고 글 작성 스케쥴을 생성하는 모델들을 구성했고, 생성한 시각에 맞춰 모델들에 뉴스 제목을 입력해 댓글을 출력 받고 트위터에 작성하는 프로그램을 구현했다. 본 연구 결과는 온라인 게임 커뮤니티 활성화, Q&A 서비스 등에 적용이 가능할 것으로 예상된다.

ABSTRACT

Many studies in deep learning show results as good as human's decision in various fields. And importance of activation of online-community and SNS grows up in game industry. Even it decides whether a game can be successful or not. The purpose of this study is to construct a system which can read texts and create comments according to schedule in online-community and SNS using deep learning. Using recurrent neural network, we constructed models generating a comment and a schedule of writing comments, and made program choosing a news title and uploading the comment at twitter in calculated time automatically. This study can be applied to activating an online game community, a Q&A service, etc.

Keywords : Deep Learning(딥 러닝), Online Community(온라인 커뮤니티), Natural Language Generation(자연어 생성)

Received: Sep. 10. 2018 Revised: Oct. 5. 2018
Accepted: Oct. 8. 2018
Corresponding Author: Kyoung-chul Kim (Korea Polytechnic University)
E-mail: ken@kpu.ac.kr

ISSN: 1598-4540 / eISSN: 2287-8211

1. 서 론

최근 많은 분야에서 인공 신경망(Neural Network)을 이용한 관련 연구들이 사람을 대체할 수 있을 정도의 성과를 보이고 있다. 인공 신경망과 관련된 이론들은 1940년대에 이미 등장했지만, 학습 방법의 어려움, 학습 데이터 부족, 컴퓨터 하드웨어 성능 부족 등 여러 한계가 드러나며 암흑기를 겪었다. 하지만 2006년 Geoffrey Hinton이 기존 신경망의 문제점들을 해결하는 방법을 제시했고[1], GPU를 통해 수많은 연산을 병렬로 처리할 수 있게 되면서 인공 신경망은 많은 발전을 이루었다.

2018년 5월, 구글(Google)은 새로운 인공지능 시스템 구글 듀플렉스(Google Duplex)를 소개했다[2]. 구글이 공개한 데모 동영상에서 이 시스템이 탑재된 인공지능 비서 구글 어시스턴트(Google Assistant)가 미용실에 예약을 하기 위해 전화를 거는 모습이 나왔다. 구글 어시스턴트는 직원의 말을 이해하고 자연스럽게 대답하며 대화를 이어갔고 미용 시간을 예약 했다. 게임 분야에서도 딥 러닝을 이용한 프로젝트가 이슈가 된 적이 있다. 비영리 인공지능 연구 기업 OpenAI가 딥 러닝을 이용해 만든 인공지능이 2017년에 온라인 게임 Dota2의 프로게이머들과의 1대1 대결에서 승리했다[3]. 이 인공지능은 다양한 변수가 존재하는 복잡한 상황에서 실시간으로 판단해 행동했고 선수들을 상대로 여러 차례 승리했다. 이처럼 여러 분야에서 딥 러닝을 접목한 연구들이 이루어지고 있고, 실생활에도 많이 적용되고 있다.

또한 게임 산업에서 SNS, 게임 관련 커뮤니티의 활성화는 중요한 요소이다. 만약 이런 공간에서 다른 사용자들과 문답을 통해 정보를 공유하고, 사용자의 글에 반응을 하는 등의 활동을 할 수 있는 인공지능 프로그램이 있다면 커뮤니티에서 정보 제공의 역할과 소통의 역할을 대신 할 수 있을 것이다.

이에 본 연구는 딥 러닝을 통해 SNS, 온라인 커뮤니티 사용자들이 작성한 텍스트에 대한 반응들을 보고 학습해 그 활동 패턴을 모사하는 시스템을 구성하는 것을 목표로 한다. 텍스트는 길이가 가변적이고 연속성이 있는 데이터이기 때문에 이런 데이터를 다루기 적합한 순환 신경망을 연구에 사용했다.

2. 딥 러닝

딥 러닝 분야에서 가장 많이 사용되는 신경망 구조로 합성곱 신경망(Convolutional Neural Network, CNN)과 순환 신경망(Recurrent Neural Network, RNN)이 있다.

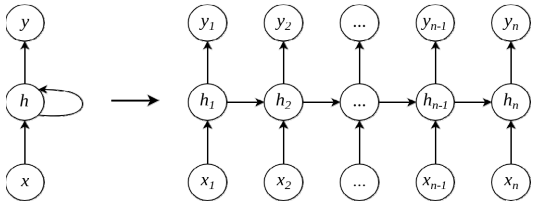
CNN은 주로 2D 이미지를 입력으로 받아서 처리하는 신경망이다. CNN은 합성곱 계층(Convolution Layer)과 풀링 계층(Pooling Layer)로 이루어지는데, 합성곱 계층에서는 이미지에 필터들을 적용해서 특징들을 추출해내고, 풀링 계층은 이 특징들에서 값을 샘플링해서 이미지 내 특징의 상대 위치에 둔감해지도록 만든다. 마지막으로 이 뒤에 최종 판단을 위한 Fully-Connected Layer가 추가된다. CNN은 이미지에서 특정 패턴을 분류하는데 많이 사용된다. ImageNet 내의 130만개의 고해상도 이미지들을 CNN을 이용해 1000가지의 종류로 분류한 연구가 존재하고[4] CNN을 도입해 빠르게 얼굴 인식을 가능하게 한 연구도 존재한다[5].

이와 달리 RNN은 주로 연속적인 흐름이 있는 데이터를 입력으로 받는다. RNN은 이전 입력들을 저장해 놓은 상태를 다음 단계로 전달해서 다음 단계의 새로운 입력과 같이 사용한다. 이를 통해 연속적인 데이터의 처리가 가능하다. RNN과 CNN을 연결해 그림을 입력 받아 인식하고 그에 대한 설명을 출력하는 연구도 있다[6].

본 연구에서는 SNS에 글을 작성하는 것을 목표로 하기 때문에 RNN을 이용해 연구를 진행했다.

2.1 순환 신경망

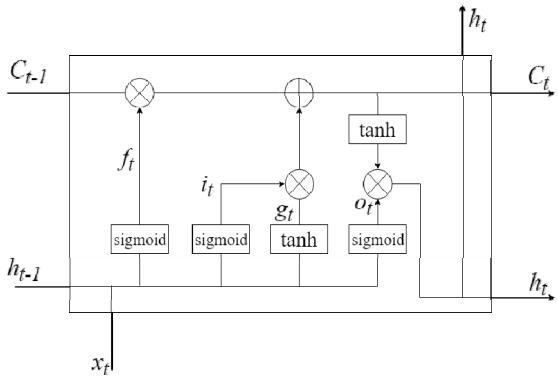
일반적인 신경망은 가중치 값을 갖는 노드들로 이루어져 있고 그 노드들은 새로운 입력에만 대응하는 가중치 값을 가진다. 그러므로 이전의 입력 값은 반영하지 않고 새로운 입력 값을 이용해 계산한 오차에 따라서 가중치를 갱신한다. 이와 달리 RNN은 노드의 가중치 값들뿐만 아니라 추가적으로 이전 입력 값들을 합쳐 저장한 상태를 내부에 갖는다. 또한 노드들은 저장된 상태 값에 대응하는 가중치 값도 포함한다. 이런 변수들이 RNN에 추가된 이유는 시계열 데이터를 처리하기 위함이다. 시계열 데이터란 시간이 흐름에 따라 값이 연속적으로 변하는 데이터를 말한다. 예를 들어 주가는 시간에 따라 값이 계속해서 변하므로 시계열 데이터라고 할 수 있다. RNN 노드의 상태 값은 시계열 데이터를 처리하면서 마치 사람의 기억력과 같은 역할을 한다. 매 입력 단계마다 RNN 노드는 이전 단계의 상태와 현재의 입력을 조합해서 새로운 상태 값을 만들어 저장한다. 이렇게 함으로써 RNN은 이전 입력과 새로운 값 사이의 연관성을 파악할 수 있게 된다[7].



[Fig. 1] A recurrent network with no outputs

[Fig. 1]에서 볼 수 있듯이 RNN은 이전의 입력에 의해 만들어진 상태와 새로운 입력을 같이 사용해 결과를 생성한다. RNN은 연속적인 데이터 처리가 가능하기 때문에 동영상이나 텍스트 관련 연구에서 좋은 성능을 보인다. RNN을 이용해 음성인식 성능을 높이려고 시도한 연구가 있고[8], 뼈대 기반 동작 인식을 구현한 연구도 있다[9].

본 연구에서는 3가지의 RNN 기반 모델을 사용해 시스템을 구성했다. 그리고 각 모델을 구성하기 위해 Long Short-Term Memory (LSTM)[10] cell 신경망이 이용됐다.



[Fig. 2] Long Short-Term Memory Cell

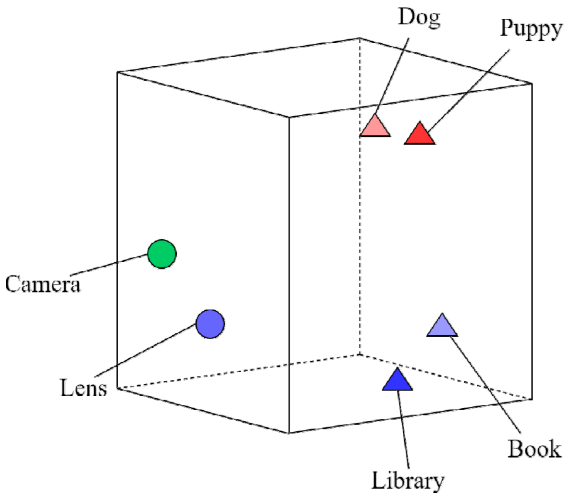
LSTM은 이전 RNN Cell인 Vanilla RNN Cell을 보완하기 위해 나온 모델이다. 기존 Vanilla RNN Cell은 오래전 과거의 입력까지 상태에 계속 저장하면서 곱해나간다. 이럴 경우 기대 값에 대한 결과 값의 오차를 줄이는 역전파(Back Propagation) 과정에서 오차 변화량이 너무 작아지거나 커지는 Gradient Vanishing, Gradient Exploding 문제가 발생한다. 이 문제에 영향을 덜 받기 위해 LSTM에는 Cell State와 여러 Gate들이 추가되었다. Cell State는 Hidden State와 별개로 이전 상태들의 값을 다음 단계로 전달하는 역할을 수행한다. Cell State를 갱신하는 연산엔 덧셈이 추가되어 있기 때문에 기존의 Vanilla RNN Cell보다 Gradient Vanishing 등의 문제가 덜 일어나게 된다. Gate들은 이전 단계의 Hidden State와 현재 입력을 이용해서 Cell State와 Hidden State를 갱신하는 역할을 맡는다. Forget Gate는 기존의 정보를 Cell State에서 얼마나 잊을지 결정하고, Input Gate는 새로운 정보를 Cell State에 얼마나 추가할지 결정한다. 마지막으로 Output Gate는 새롭게 갱신된 Cell State에서 어떤 부분을 새로운 Hidden State로 출력할 것인지 결정한다.

구성한 세 개의 모델 중 Char-RNN, 작성 스케쥴 생성 모델은 LSTM 신경망을 여러 층으로 조

합해 구성했다. 또 다른 모델인 NMT 모델은 앞 모델들의 구조 두 개를 조합해 구성됐다.

2.2 워드 임베딩

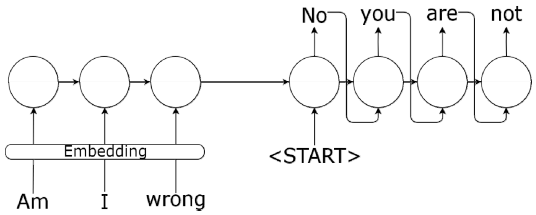
특정 단어나 글자가 어떤 의미를 갖는지 모델이 이해하도록 하기 위해선 단어나 글자를 벡터로 나타낼 필요가 있다. 이때 각 단어가 어떤 값을 가질지를 모델과 같이 학습시키는데, 이것을 워드 임베딩(Word Embedding)이라고 부른다[11]. 본 연구에서 구성한 모델들은 단어를 워드 임베딩을 통해 벡터로 변환시킨 후 입력으로 사용한다.



[Fig. 3] 3D Word Embedding

2.3 시퀀스-투-시퀀스 모델

시퀀스-투-시퀀스(Sequence-to-Sequence) 모델은 시퀀스를 입력으로 받아서 시퀀스를 생성해 출력하는 모델로, 입력 시퀀스 데이터를 처리하는 인코더(Encoder)와 출력 시퀀스를 처리하는 디코더(Decoder)로 구성된다. 인코더와 디코더는 각각 하나의 RNN모델이며 이를 하나로 합친 게 시퀀스-투-시퀀스 모델이다[12].



[Fig. 4] Example of an encoder-decoder or sequence-to-sequence RNN architecture

인코더에서는 문장의 단어들을 하나씩 입력 받아 문장을 숫자 행렬로 나타내는 인코딩(Encoding)작업을 한다. 이 작업을 통해 문장의 상태를 나타내는 값을 얻을 수 있다. 디코더에서는 인코더의 결과 값을 이용해 이 작업이 역으로 수행된다. 이때 인코더가 문장을 상태로 만드는 과정과 디코더가 전달된 입력 상태를 문장으로 바꾸는 과정을 입력 값과 결과 값에 맞게 학습시킨다. 시퀀스-투-시퀀스 모델을 이용해 문장을 다른 언어로 번역하는 연구가 존재하는데[13], 기존의 확률 기반 번역보다 더 높은 성능을 보였다. 또한 영상을 입력 받아 그에 대한 설명을 작성하는 연구도 존재한다[14].

3. 연구 방법

3.1 학습 데이터 종류 및 수집

사람이 작성한 것과 유사한 행태로 댓글을 생성하려면, 1)어떤 내용의 댓글을 생성할지, 2)언제 그 댓글을 올릴지를 결정해야 한다. 이를 위하여 많은 사람들의 공감을 받은 비교적 검증된 댓글인 포탈의 상위권 댓글을 학습 데이터로 이용했다. '네이버(Naver)'는 정치 분야 뉴스에서 '공감순'을 제거했기 때문에 그 다음으로 이용자가 많은 '다음(Daum)'에서 데이터를 수집했다. 이를 위해 웹 브라우저 자동화 툴인 Selenium을 이용해 연구기간 동안 매일 전날의 댓글 많은 상위 50위 뉴스들의 댓글들과 뉴스 데이터를 수집했다.

[Fig. 5] An example of Text Data

또한 댓글을 작성할 시각을 모방하기 위한 학습 데이터도 같이 수집했다. '다음'에서는 특정 아이디가 작성한 댓글들을 볼 수 있도록 제공하는데, 위 댓글 많은 뉴스들에 있는 공감순 상위 3개 댓글 작성자들의 최근 6개월 동안 댓글 작성 시각들을 수집했다.

[Fig. 6] An example of Schedule Generation Model's Training Data

데이터 수집은 2018년 2월 1일 부터 2018년 8월 19일 까지 수행되었으며, 약 9300개 뉴스의 데이터를 수집했다.

3.2 모델 학습

본 연구를 위해 총 세 가지 모델을 사용했으며 모두 Tensorflow 1.9.0-dev20180501 프레임워크와 Python 3.5.2 언어를 이용해 구성했다. CPU는 Intel Core i7-7700 3.60GHz, RAM 64G, GPU NVIDIA GTX 1080ti 2개를 이용한 하드웨어 환경에서 실험을 진행했다. 텍스트 생성을 위한 모델 Char-RNN, NMT, 작성 스케쥴 생성을 위한 모델이 존재한다.

비교적 많은 사람들이 공감하는 댓글들을 학습시키기 위해 수집한 댓글들에 공감수에서 비공감수의 2배를 뺀 것을 점수로 매기고 그 내림차순으로 정렬한 뒤, 상위 10%를 모델 학습용 데이터로 추출했다. 이렇게 추출된 데이터 중 70%는 실제 학습에 사용했고, 나머지 30%의 데이터는 모델의 학습 정도를 확인하기 위한 테스트 데이터 셋으로 활용했다.

텍스트 생성을 위해 Char-RNN, NMT 두 가지 모델을 구성했다. Char-RNN 모델은 글자를 하나씩 출력할 때마다 이전 문장을 반영해 예측하는 모델이다. NMT는 입력 문장의 상태를 먼저 생성 후 그것을 새로 해독하는 모델이다. 각 모델의 텍스트 생성 방식이 다르기 때문에 두 가지를 모두 사용했다.

3.2.1 Char-RNN

Char-RNN은 문장 안의 문자들을 시계열 데이터로 보고 학습하는 RNN 모델이다. 문장이 입력되었을 때 특정 글자 다음에 어떤 글자가 나타났는지 보고 그 패턴을 학습한다. 이 모델은 스탠포드 대학의 Andrej Karpathy가 제안한 것으로[15], 입력된 문장들과 유사한 문장들을 생성하도록 신경망을 학습시킬 수 있다. 텍스트를 다루는 RNN 모델 중 가장 기본적인 모델로 다른 RNN 모델의 성능을 측정하기 위한 비교 모델로 이용하기도 한다.

Karpathy가 직접 구현한 Char-RNN은 Lua언어로 구현 했고 Torch 프레임 워크를 사용했다[16]. 본 연구에서는 이를 Python과 Tensorflow 프레임워크를 사용해 재구성한 오픈소스 프로젝트[17]를 이용해 모델을 구성했다.

이 모델은 1024개의 상태 값을 가지는 LSTM 셀로 구성한 3개의 레이어로 이루어져 있다. 모델의 학습률은 0.01, Sequence Length는 100으로 설

정했다. 추가로 연결된 기본적인 뉴럴 네트워크 구조인 Fully-Connected Network에서 RNN 모델의 출력 값을 이용해 다음 글자가 어떤 것인지 최종적으로 판단한다.

입력 데이터들은 기사의 제목과 그 기사에 대한 댓글이 탭(tab) 문자로 연결되어 있고 개행 문자로 끝나는 구조로 되어있다. 입력 데이터를 32개씩 묶고, 전체 입력에 대한 Batch Data들을 100회 반복해서 학습한다.

처음 학습 시 입력 데이터에 있는 모든 글자를 Vocabulary에 저장하고 워드 임베딩 데이터를 같이 학습한다. 이 때문에 새로운 데이터로 학습 시 기존 Vocabulary에 없는 글자가 나오면 워드 임베딩 범위 초과 오류가 발생할 수 있다. 이를 대비해 새로운 학습 데이터에서 Vocabulary에 없는 글자는 제외하도록 했다. 이때 제외되는 글자는 대부분 잘 사용되지 않는 특수문자나 한자로 나타났다.

학습된 모델을 통해 샘플링을 할 땐 Prime Text에 기사 제목과 탭 문자를 주면 모델이 댓글에 해당하는 문장을 생성해서 출력한다. 제목의 글자들이 모델에 입력될 때마다 모델의 상태가 갱신된다. 글자들이 처리된 다음엔 문장의 끝을 의미하는 개행 문자가 나올 때까지 반복해서 다음 글자를 생성해 출력한다.

3.2.2 Neural Machine Translation

시퀀스-투-시퀀스 구조로 구현된 기계번역(Neural Machine Translation, NMT)모델로 구글에서 제공하는 NMT 오픈소스 프로젝트[18]를 이용해 구성했다. 입력 값은 기사 제목, 출력 값은 댓글로 설정하고 모델을 학습시켰다. Encoder와 Decoder 모두 128개의 상태 값은 갖는 LSTM Cell을 2개 레이어로 연결해 구성했다. 학습 데이터에서 제목과 댓글 쌍을 임의로 12000번 선택해 모델을 학습시켰다.

NMT 모델은 Char-RNN 모델과 다르게 글자 단위가 아닌 공백으로 구분되는 단어를 단위로 학습한다. 한글은 조사가 다양하게 사용되고 띄어쓰기가 제대로 사용되지 않아도 이해하는데 크게 지장이 없기 때문에 공백을 단위로 단어를 자르면 단어 수가 영어에 비해 훨씬 많아진다. 때문에 한글 텍스트를 NMT 모델에 학습시킬 때 메모리가 부족해 진행할 수 없는 문제가 빈번하게 발생했다. 때문에 Char-RNN 모델과 다르게 NMT모델을 학습할 땐 더 적은 뉴스 데이터에 대해, 그 댓글들의 상위 10% 중 7000개만 학습하도록 전처리를 했다.

3.2.3 작성 스케쥴 생성 모델

구성한 시스템이 SNS에서 사람처럼 활동하기 위해 언제 글을 작성할지에 대한 학습도 필요하다. 따라서 포탈 이용자들의 댓글 작성시각을 수집하고 이를 학습할 댓글 작성 스케쥴 생성 모델을 구현했다. 시각을 수집 할 때 댓글이 작성된 시각을 유닉스 시간(1970년 1월 1일부터 해당 시각까지 몇 초가 흘렀는지를 나타내는 정수) 형식으로 변환해 저장했다. 2018년 6월 17일부터 동년 7월 14일까지 약 212만개의 시각 데이터를 수집했다.

이때 본 연구에 필요한 생성 시각들은 하루 동안 언제 글을 작성해야 하는지 이다. 때문에 수집한 시각 데이터의 년, 월, 일은 삭제하고 분, 초를 기반으로 데이터를 변환했다. 수집한 시각 데이터들을 작성한 당일 0시 정각을 기준으로 몇 초가 지났는지 나타내는 정수로 변환해 학습 데이터로 이용했다.

작성 스케쥴 생성 모델은 128개의 상태 값을 갖는 LSTM Cell을 3개의 레이어로 쌓아 구현했다. 추가로 RNN 모델 뒤에 Fully-Connected Network를 연결해 최종 예측 시각을 계산하도록 구현했다.

3.3 자동화 프로그램 구현

SNS 상에서의 활동을 위해 학습, 결과 생성, 업로드 작업들을 자동화했다. 학습 데이터를 모으고, 수집한 데이터로 모델들을 학습하고, 결과를 출력

해 SNS에 작성하는 과정을 스크립트로 만들어 매일 동작하도록 구현했다.

스크립트가 오전 0시 1분에 실행되면 전날의 댓글이 가장 많은 50개의 뉴스와 그 댓글들을 수집한다. 수집이 끝나면 저장된 댓글들을 필터링 하고 Char-RNN모델과 NMT모델에 입력으로 넣어 각 모델을 학습한다. 두 모델이 학습을 완료하는 데에는 위에 기술한 사양의 서버에서 약 6시간이 소요되었다.

위의 학습 과정이 끝나면 작성 스케쥴 생성 모델이 직전 글이 작성 된 시각을 입력으로 받아 글을 작성할 스케쥴을 생성해낸다. 그날 글을 작성할 스케쥴을 모델로 부터 출력 받고, 생성된 시각이 될 때마다 Daum 포탈의 메인 화면 뉴스 중 상위 5개를 추출한다. 그 중 임의로 하나를 선택하고 선택된 뉴스 제목을 Char-RNN, NMT 모델에 입력하고 각 모델로부터 댓글을 출력 받는다. 구성한 자동화 프로그램이 댓글을 직접 포탈에 작성하면서 활동하면 불법이기 때문에 트위터에 작성했다. 따라서 자동화 프로그램이 선택된 뉴스의 제목과 URL 주소, 각 모델이 출력한 댓글을 트위터 API를 이용해 트위터에 작성하도록 구현했다.

4. 연구 결과

각 뉴스 제목에 대한 텍스트 생성 모델의 결과는 다음의 [Table 1], [Table 2], [Table 3]과 같다.

[Table 1] Results of Text Models

News Title	이재명 과거 '강제입원 의혹' 논란 다시 불거져
Char-RNN	뽑네 다른 딸은 무슨 치명성을 이야기하나 . . .
NMT	진짜 요즘 이재명 죽이기 ~ ~ ! ! ! !

[Table 2] Results of Text Models

News Title	[종합2보]신일그룹 "돈스코이호 보물, 현재 파악할 수 없는 상황"
Char-RNN	국민들 세금을 지켜야 한다 . 그리고 또 하는 거 , 절대 아닌 정치인은 없을거다 . . .
NMT	아무리 생각해도 ㅁ ㅊ ㅊ

[Table 3] Results of Text Models

News Title	'국정원 특활비' 박근혜도 뇌물 인정 안돼..MB에도 영향 줄듯
Char-RNN	아닐때 잡아가시갈.
NMT	이런 개 . . . ㅜ

Char-RNN의 경우 글자 기반 모델이기 때문에 문장에 의미 없는 경우가 많지만, 일반적 댓글 문체와 유사한 글이 생성된다. NMT는 단어 기반 모델이며 Char-RNN과 비교했을 때 뉴스 제목과 결과의 연관성이 높고 이해 가능한 텍스트들이 생성됐다.

작성 스케쥴 생성 모델이 생성한 날짜 별 시각은 다음과 같다.

[Table 4] Results of Schedule Generation Model

25th, July	AM 07:12
	PM 06:26
26th, July	PM 12:32
	PM 06:13
23th, August	PM 12:23
	PM 06:01
	PM 11:57

점심시간, 저녁시간과 같이 사람들이 여유가 있는 시각들이 생성됐다. 임의의 사람들이 작성하는 시각 데이터로 학습했기 때문에 결과물 역시 일반적인 빈도를 가지고 있다. 자주 글을 작성하는 사람들의 시각을 골라서 학습 데이터로 이용하면 더 자주 글을 작성할 수 있을 것이다.

[Fig. 7], [Fig. 8]은 실제 트위터에 작성한 결과 중 일부이다.

[Fig. 7] A Result of Automatic Generation

[Fig. 8] A Result of Automatic Generation

트위터 계정을 생성한 뒤에 다른 활동은 일체 하지 않고 자동화 스크립트를 통해 글을 생성하고 등록했다. 그런데도 일부 트윗을 리트윗(Retweet) 하는 사람들이 나타나고, 이 계정을 팔로우(Follow)하는 경우도 생겼다.

5. 결론

5.1 연구 결과 해석

모델이 생성한 텍스트의 모든 결과가 완벽하게 이해 가능한 문장은 아니지만, 충분히 자연스러운 문장들 또한 생성된다. 게다가 그렇지 못한 문장들의 단어들도 제목과 연관성이 있는 것을 볼 수 있다. 따라서 사람이 작성하는 댓글들을 딥 러닝을 이용한 모델이 모방할 수 있다고 판단된다.

작성 스케쥴 생성 모델이 생성한 시각들은 주로 아침 출근 시간인 오전 8시 전후, 점심시간인 정오 전후, 퇴근시간인 오후 6시 전후, 그리고 오후 9시에서 자정 사이로 나타난다. 이것은 일반적인 이용자들이 SNS에 글을 작성하는 패턴을 기계가 학습해 모방하고 있는 것으로 해석된다.

또한 다른 SNS 이용자들이 이번 연구의 결과물에 반응을 보인 것은 구성한 시스템과 사용자 간의 소통에 유의미한 가능성을 보여준다.

5.2 연구의 한계와 향후 연구 방향

일반적으로 영미권 대학생이 사용하는 어휘는 20000개 정도인데, 현 NMT 모델은 이에 최적화된 영어기반 모델이다. 한국어의 경우엔 맞춤법이 틀려도 소통에 지장이 없는 경우가 많아 대부분의 SNS의 글이나 커뮤니티의 댓글은 맞춤법을 지키지 않는다. 또한 단어의 뒤에 다양한 조사가 붙기 때문에 공백으로 구분한 단어의 개수가 영어에 비해 훨씬 많다. 하루 동안 수집한 댓글의 경우 공백으로 구분한 단어 수는 20000개가 넘고, 일주일의 경우 90000개가 넘었다. 어휘가 많아지면 단어 임베딩을 위한 메모리가 고갈되는 현상이 일어나 적은 양의 학습데이터로 학습 시킬 수밖에 없었다. 한국어나 일본어의 경우 이러한 문제가 많이 나타나는데, 이를 해결하기 위한 방법으로 Sub-unit[19]을 이용해 처음 보는 단어를 처리하는 방법이 있다. 이밖에 시퀀스-투-시퀀스 모델에 Attention Mechanism[20]을 도입하는 등 관련 연

구가 활발히 진행되고 있어 이러한 발전 상황들을 모델에 적용하면 더 높은 성능을 보여줄 것으로 기대된다. 향후 본 연구의 모델을 보완해 게임 커뮤니티에 활용하면 해당 사용자들에게 정보제공이나 소통 측면에 긍정적인 영향을 줄 수 있을 것으로 예상된다.

이번 연구에서 사용자들의 반응을 분석하는 것에 대한 어려움도 있었다. 구성한 시스템이 직접 댓글을 올리면서 활동하는 것은 불법이기 때문에 생성한 댓글을 뉴스에 직접 등록하지 못했다. 따라서 다른 이용자들의 반응을 살피는 데 한계가 있었다.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea Grant funded by Korean Govenment (NRF-2017R1A2B1009495)

REFERENCES

[1] Geoffrey E. Hinton , Simon Osindero and Yee-Whye Teh, “A Fast Learning Algorithm for Deep Belief Nets”, Neural Computation Volume 18 Issue 7, pp.1527-1554, 2006.

[2] Google, “Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone”, https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html, 2018.

[3] OpenAI, “Dota 2”, https://blog.openai.com/dota-2/, 2017.

[4] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks, 2012.

[5] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, Gang Hua, “A Convolutional Neural Network Cascade for Face Detection”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.5325-5334, 2015.

[6] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, Alan Yuille “Deep Captioning With Multimodal Recurrent Neural Networks (m-RNN)”, https://arxiv.org/abs/1412.6632, 2015.

[7] Ian Goodfellow, Yoshua Bengio And Aron Courville, “Deep Learning”, The MIT Press, pp.363-382, 2016.

[8] Alex Graves, Abdel-rahman Mohamed, Geoffrey Hinton, “Speech Recognition with Deep Recurrent Neural Networks”, https://arxiv.org/abs/1303.5778, 2013.

[9] Yong Du, Wei Wang, Liang Wang, “Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1110-1118, 2015.

[10] “Long Short-Term Memory”, Neural Computation Volume 9 Issue 8, pp.1735-1780, 1997.

[11] Oren Melamud, Omer Levy, Ido Dagan, “A Simple Word Embedding Model for Lexical Substitution”, Proceedings of NAACL-HLT 2015, pp.1-7, 2015.

[12] Ian Goodfellow, Yoshua Bengio And Aron Courville, “Deep Learning”, The MIT Press, pp.385-400, 2016.

[13] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, Yoshua Bengio, “On the properties of neural machine translation: Encoder-decoder approaches”, https://arxiv.org/abs/1409.1259, 2014.

[14] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, Kate Saenko, “Sequence to Sequence - Video to Text”, The IEEE International Conference on Computer Vision (ICCV), pp.4534-4542, 2015

[15] Andrej Karpathy, “The Unreasonable Effectiveness of Recurrent Neural Networks”, http://karpathy.github.io/2015/05/21/rnn-effectiveness, 2015.

[16] Andrej Karpathy, “Multi-layer Recurrent Neural Networks (LSTM, GRU, RNN) for character-level language models in Torch” https://github.com/karpathy/char-rnn, 2015.

[17] insikk, “Korean language requires a little different treatment when we run character level RNN”,

https://github.com/insikk/kor-char-rnn-tensorflow, 2017.
[18] Google, “TensorFlow Neural Machine Translation Tutorial”, https://github.com/tensorflow/nmt, 2017.
[19] Rico Sennrich, Barry Haddow, Alexandra Birch, “Neural Machine Translation of Rare Words with Subword Units”, https://arxiv.org/abs/1508.07909, 2016.
[20] Minh-Thang Luong, Hieu Pham, Christopher D. Manning, “Effective Approaches to Attention-based Neural Machine Translation”, https://arxiv.org/abs/1508.04025, 2015.

최 재 용 (Choi, Jae Yong)

약 력 : 한국산업기술대학교 게임공학과 학사과정

관심분야 : 게임 프로그래밍, AI

성 소 윤 (Sung, So Yun)

약 력 : 한국산업기술대학교 게임공학과 학사과정

관심분야 : 기계학습, 게임 프로그래밍

김 경 철 (Kim, Kyoung Chul)

약 력 : KAIST 과학기술대학 전산학과 학사
KAIST 정보및통신공학과 석사
KAIST 전산학과 박사
㈜고누소프트 가약스부문 차장
현 한국산업기술대학교 게임공학부 부교수

관심분야 :컴퓨터구조, 분산처리, 온라인 게임 서버