Journal of Korea Game Society 2019 Oct; 19(5): 103-114
http://dx.doi.org/10.7583/JKGS.2019.19.5.103

인공지능

트랜스포머를 이용한 향상된 댓글 생성에 관한 연구

성소윤, 최재용, 김경철
한국산업기술대학교 게임공학과
{tjdthdbs12, jayong93, ken}@kpu.ac.kr

A Study on Improved Comments Generation Using Transformer

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

요 약

온라인 커뮤니티 안에서 다른 사용자들의 글에 반응할 수 있는 딥러닝 연구를 2017년부터 진행해 왔으나, 한국어의 조사와 같은 특성으로 인한 단어처리의 어려움과 RNN 모델의 특성으로 인한 GPU 사용률 저조 문제로 인해 적은 양의 데이터로 학습을 제한해야 했다. 하지만 최근 자연어 처리 분야의 급격한 발전으로 이전보다 뛰어난 모델들이 등장함에 따라 본 연구에서는 이러한 발전된 모델을 적용해 더 나은 학습 결과를 생성해 내는 것을 목표로 한다. 이를 위해 셀프-어텐션 개념이 적용된 트랜스포머모델을 도입했고 여기에 한국어 형태소 분석기 MeCab을 적용해 단어처리의 어려움을 완화했다.

ABSTRACT

We have been studying a deep-learning program that can communicate with other users in online communities since 2017. But there were problems with processing a Korean data set because of Korean characteristics. Also, low usage of GPUs of RNN models was a problem too. In this study, as Natural Language Processing models are improved, we aim to make better results using these improved models. To archive this, we use a Transformer model which includes Self-Attention mechanism. Also we use MeCab, korean morphological analyzer, to address a problem with processing korean words.

Keywords : Deep Learning(딥 러닝), Natural Language Processing(자연어 처리), Self-Attention(셀프-어텐션), Transformer(트랜스포머)

Received: Sep. 09. 2019 Revised: Sep. 30. 2019
Accepted: Oct. 14. 2019
Corresponding Author: Kyoung-chul Kim (Korea Polytechnic University)
E-mail: ken@kpu.ac.kr

ISSN: 1598-4540 / eISSN: 2287-8211

1. 서 론

SNS와 온라인 커뮤니티가 발전하고 활성화 되면서 이러한 가상공간에서 이루어지는 사람들의 평가에 따라 게임의 흥행이 결정되는 경우가 많아지고 있다. 대표적인 게임 커뮤니티 '루리웹'에서는 PC, 온라인 게임뿐만 아니라 다양한 콘솔 게임에 대한 평가와 추천이 이루어지는데, 기존 유저들의 반응이 신규 유저들에게 게임을 선택하는 가이드라인 역할을 하고 있다. 이러한 공간들을 통해 정보를 제공하고 다른 사용자들의 말에 반응 할 수 있는 인공지능 프로그램이 있다면 커뮤니티 활성화에 유용할 것이라 생각해 2017년부터 포털에서 제공하는 뉴스에 대한 댓글들을 수집해 학습하고 댓글 작성 활동을 모방할 수 있는 딥러닝 프로그램 연구를 진행해 오고 있다[1].

그러나 우리의 이전 연구에는 몇 가지 문제점이 있었다. 우선 모델을 학습할 때 필요한 메모리 사용량이 매우 많았다. 이 문제는 기존 시스템의 단어를 구분하는 방식과 한국어의 특성 때문에 일어났다. 기존의 모델들은 영어를 사용할 것으로 가정하고 설계되었다[2]. 영어는 문장을 공백을 기준으로 단어를 분리하더라도 평균적으로 사용하는 전체 어휘의 수가 20000여개 정도이다. 그에 반해 한글은 조사가 한 단어 안에 같이 있기 때문에 공백으로 단어를 분리하면 사용되는 어휘의 수가 급격히 늘어나게 된다. 이 문제를 해결하지 않고 위 모델들에 한국어 문장을 바로 학습 데이터로 입력하면 각 단어의 의미를 수치로 벡터화 하는 워드임베딩(Embedding) 과정에서 사용되는 메모리가 급격하게 늘어나고, 가용 메모리를 초과하면서 메모리 오류가 일어나며 학습이 중단됐다. 때문에 적은 양의 학습 데이터를 입력으로 줄 수밖에 없었다. 게다가 기존 RNN(Recurrent Neural Network, 순환신경망) 기반 모델들은 병렬성이 다른 모델들에 비해 낮기 때문에 학습 속도가 빠르지 않았다[3]. 메모리 사용량 문제를 해결하더라도 학습 데이터양이 많아지기 때문에 학습 속도가 크게 떨어지는 문제가 있었다.

하지만 지난 2년간 NLP(Neural Language Processing, 자연어 처리) 분야에서 많은 연구가 이루어졌고 인상적인 성과들을 보여주었다. 입력의 어느 곳이 출력에 밀접한 영향을 주는지를 학습을 통해 구해보려는 어텐션의 개념과[4], 그 어텐션 매커니즘을 이용하여 기존 LSTM(Long-Short Term Memory) 임베딩 모델을 대치하는 트랜스포머(Transformer)[5], 그리고 그 트랜스포머를 기반으로 구성된, 각종 문제에서 SOTA(state-of-the-art) 수준의 결과를 보여준 BERT(Bidirectional Encoder Representations from Transformers)[6], GPT(Generative Pre-Training)[7] 그리고 2019년에 발표된 XLNet[8], MASS(Masked Sequence to Sequence pre-training)[9] 같은 모델들 까지, 최근 몇 년간 자연어 처리 분야의 대표적인 성과라 할 수 있을 것이다. 이 모델들은 여러 테스트에서 이전의 모델들보다 뛰어난 결과를 보이고 있다.

이에 본 연구에서는 형태소 분석기를 사용해 한국어 문장들을 전처리하여 모델의 학습량을 늘릴 방법을 고안하고, 고성능의 새로운 자연어 처리 모델들을 통해 자연스러운 한국어 문장을 생성해내는 것을 목표로 한다.

2. 관련 연구

2.1 순환 신경망

일반적인 신경망이 입력층, 은닉층, 출력층까지 한 방향으로 계산 값이 흘러가는 것에 비하여, RNN은 이전에 계산된 은닉층의 결과값이 다음번 은닉층의 계산에 이용되도록 구성한 신경망 모델로, 시계열 데이터를 처리하는데 주로 사용되고 있다. 시계열 데이터란 시간 변화에 따라 값이 변화하는 데이터를 의미하며 증권시장에서 각 종목의 주가, 시간에 따라 변화하는 기상정보 등을 예로 들 수 있다. RNN은 내부에 상태값을 추가로 저장해 두었다가 새로운 입력이 들어올 때마다 상태값을 새로 갱신하면서 이

전 순서에서 처리한 입력을 보존한다[10].

그러나 이러한 구조에서는 오래전에 입력된 값에 대한 상태값이 계속 희석되기 때문에, 입력의 길이가 길어질수록 학습 능력이 떨어지는 장기의존성(longterm dependencies) 문제가 있다. 이 문제를 개선하기 위한 기존 RNN을 보완한 모델들이 존재한다. 대표적으로 기존 RNN의 은닉층과 별도로 선형계산을 통해 비교적 장기간 상태를 유지하는 셀 상태(cell state)를 추가하여 장기기억을 유지하려는 LSTM 모델[11], LSTM보다 적은수의 파라메터로 구성이 가능하여 간단하고 빠르게 계산할 수 있지만 비슷한 효과를 갖는 GRU(Gated Recurrent Unit)[12] 등이 존재한다.

이러한 RNN 모델을 이용하여 문장 번역처럼 데이터 흐름을 입력받아 새로운 데이터 흐름을 만드는 Seq2Seq(Sequence to Sequence) 모델[13]을 구성할 수 있다. Seq2Seq 모델은 인코더와 디코더가 연결된 구조로, 각각을 RNN 레이어로 구현할 수 있다. 인코더는 입력 데이터를 처리해서 특정 벡터들로 변환하고, 디코더는 인코더가 생성해낸 벡터들을 자신의 입력으로 설정해서 출력 데이터를 만든다. 우리 연구는 기사 제목을 입력으로 받아서 그 제목에 반응하는 문장을 만들어내는 것이 목적이기 때문에 Seq2Seq 모델을 주로 활용했다.

2.2 어텐션 매커니즘 (Attention Mechanism)

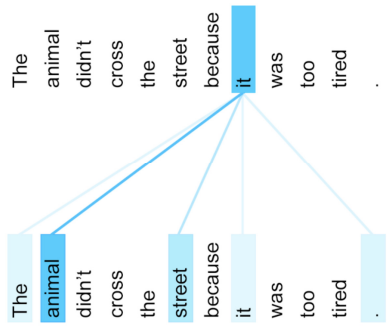
긴 시퀀스를 다루는 학습에서 장기의존성 문제가 학습 성능을 떨어뜨리는 원인이 되었다. 이를 보완하기 위해 등장한 LSTM 모델은 기존 RNN 모델들에 비해선 장기의존성 문제가 나아졌지만 여전히 이로 인한 학습 성능저하 문제를 가지고 있다.

인간은 긴 문장이나 사진을 인식할 때 요소 하나하나를 전부 인식하지 않고, 필요한 부분만을 집중해서 인식한다. 이와 유사하게 학습 시 입력 데이터 전부를 동등하게 활용하지 않고 각 부분에 가중치를 매겨 필요한 입력에만 집중할 수 있게 학습하는 것이 어텐션 매커니즘의 핵심 아이디어이다.

초기 어텐션은 LSTM 모델의 내부 상태 값들의 시퀀스에 가중치를 매기는데 사용했지만 최근에는 LSTM 모델 자체가 어텐션 모델들로 대체되고 있다. 이렇게 인코더와 디코더를 어텐션 레이어 로만 구성한 모델들이 기존 모델들보다 뛰어난 성능을 보이고 있다. 뿐만 아니라 RNN 모델들은 시퀀스가 전부 입력되어야 back propagation through time(BPTT)로 학습할 수 있는데 비해 어텐션 모델들은 병렬성이 높아 같은 시간 동안 더 많은 양의 데이터를 학습할 수 있다.

2.3 트랜스포머(Transformer)

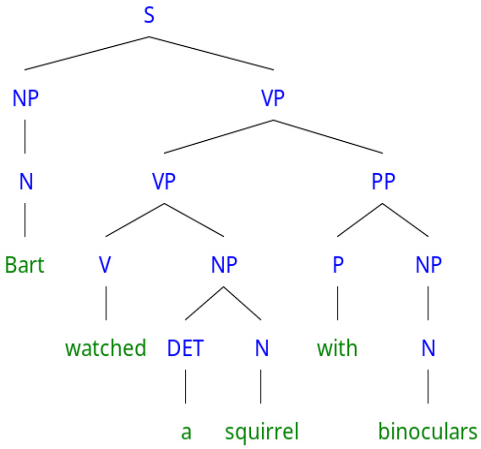
문장 안에서 어떤 단어의 의미를 제대로 파악하기 위해선 문장의 앞부분뿐만 아니라 뒷부분 까지 읽으며 문맥을 파악한다. 예를 들어 “넌 죽을 준비해… 난 밥을 준비할 테니” 와 같은 문장에서 ‘죽을’이란 단어의 의미를 앞부분 ‘넌’만 가지고 정확한 의미를 파악하기 힘들다. 기존 LSTM 모델을 통한 임베딩은 입력을 순차적으로 처리하기 때문에 앞의 입력들만을 이용해 현재 입력에 대한 상태를 생성한다. 이를 보완하기 위해 bidirectional LSTM 모델[14]과 셀프-에텐션(self-attention) 모델[15]이 존재한다. 트랜스포머는 이중 셀프-어텐션 모델을 이용했다. 셀프-어텐션 모델은 입력을 한 번에 받아 입력 안의 단어들 간의 관계를 파악한다. 이를 통해 문장 안에서 어떤 한 단어의 의미를 다른 단어들과의 연관성 가중치 정도를 이용해 판단할 수 있다.



[Fig. 1] An example diagram for attention mechanism[16]

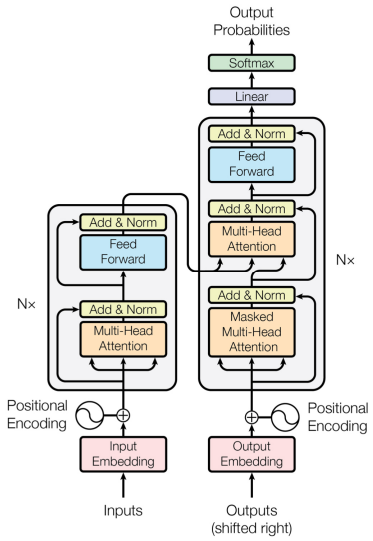
[Fig. 1]에서의 'it'의 의미를 파악할 때 셀프-어텐션 모델을 통해 다른 입력들에 대한 각각의 연관성 가중치를 매길 수 있다. 이때 가장 높은 가중치를 가진 입력 'animal'을 통해 'it'의 의미를 파악할 수 있다. 이는 또한 기존 워드 임베딩의 한계였던 동음이의어의 의미 파악에도 활용될 수 있다.

셀프-어텐션은 이처럼 단어 사이의 관계를 계산하는데 이때 여러 레이어와 헤드에서 이러한 계산 결과가 존재한다. 이러한 여러 계층의 계산 결과들을, [Fig. 2]처럼 문장을 구조화 하는 여러 계층을 가진 구문 트리(syntax tree)로 간주할 수 있다고 해석한 시도도 있다.[17]



[Fig. 2] An example of parsing tree[17]

트랜스포머는 기존 RNN 모델 레이어들로 구성된 Seq2Seq 모델을 셀프-어텐션 레이어로 구성한 모델이다. 트랜스포머 모델은 [Fig. 3]와 같이 셀프-어텐션 레이어와 feed forward 레이어로 구성된 6개의 인코더들과 셀프-어텐션, feed forward 레이어에 인코더-디코더 어텐션을 더한 디코더들로 구성된다.



[Fig. 3] A structure of Transformer model[18]

그러나 이러한 셀프-어텐션 모델은 기존 RNN 모델에 비해 시퀀스 안에서의 입력 순서를 감안하지 못하는 문제가 있다. 따라서 이를 보완하기 위해 위치 인코딩(positional encoding)이 사용됐다. 이는 각 입력의 순서에 따라 생성한 위치 벡터 값을 더하는 방식으로 이루어진다.

비슷한 시기에 진행된 유사 연구로 Hacker News 기사와 댓글을 트랜스포머 모델로 학습한 'hncynic' 프로젝트[19]가 있다. 그러나 본 연구와는 몇 가지 차이가 있다. 우선 위 연구에서는 Hacker News에서 제공하는 기사들을 특정 시점에 한번 덤프(dump)한 데이터를 이용해서 모델을 학습하는 반면, 우리는 웹 크롤러를 구현하여 새로운 다음 뉴스와 댓글들을 매일 직접 수집하고 이를 이용해 학습한다. 이렇게 함으로 우리의 모델은 새로운 주제에 대한 뉴스와 댓글들에 대응할 수 있게 된다. 또한 데이터의 전처리에서도 차이점이 있는데 'hncynic' 프로젝트에서는 각 단어를 BPE(Byte Pair Encoding)[20]을 사용해서 분해하고, 우리는 형태소 분석기 MeCab를 사용해서 각 단어가 가지는 의미에 따라 분해한다. 또한 수집한 댓글 중에서 이용자들의 공감 수가 비공감 수의 2배 보다 많은 순으로 상위 10%의 댓글만을 골라

내어 학습하는 본 연구와 다르게 'hncynic' 프로젝트는 모든 댓글을 학습에 사용한다.

2.4 최근의 언어 모델

최근에 자연어 처리 분야에서 트랜스포머 구조를 이용해 더욱 성능을 높인 새로운 모델들이 나타나고 있다. 대표적으로 구글의 BERT가 그 중 하나이다. BERT는 트랜스포머의 인코더 부분만을 이용해 입력 시퀀스를 벡터들로 표현하는 모델이다. BERT는 원하는 목적의 데이터로 학습하기 전에 라벨링 되지 않은 문장 데이터들을 이용해 미리 언어 모델을 학습(Pre-training)시키고, 그 후에 원하는 목적의 데이터로 모델을 최적화 시키는 방식으로 학습된다. BERT는 사전 학습 단계에서 입력들 중 일부를 생략하고(Masking), 생략된 단어를 모델이 예측하는 방식으로 학습시켜 단어가 포함된 문맥에 따라 그 단어의 임베딩 값을 변화시키도록 만들었다. 이 모델을 SQuAD(The Stanford Question Answering Dataset) 데이터 셋을 통해 학습시킨 결과, 최초로 자연어 처리 분야에서 인간을 뛰어 넘는 성능을 보여주었다.[21]

SQuAD 데이터 셋을 학습한 모델 중 BERT 이전에 높은 성능을 보여주었던 ELMo[22]는 BERT와 비슷하게 많은 양의 문장을 이용해 사전학습을 거쳤으나, BERT와는 다르게 사전 학습 방식이 간소했고, bidirectional LSTM 기반으로 상대적으로 낮은 성능을 보여주었다.

BERT 이전의 또 다른 모델로 OpenAI에서 만든 GPT(Generative Pre-Training) 모델이 있다. GPT 모델은 트랜스포머를 이용해 만들어졌지만 ELMo와 비슷하게 사전 학습 방식이 간소해서 BERT와 같은 성능을 내지는 못하였다.

이후에는 BERT를 능가하는 XLNet이나 MASS 같은 모델들도 등장했다. XLNet은 구글 브레인과 카네기 멜론 대학교가 협업하여 만든 모델로 auto-regressive와 auto-encoding 방법들 중에 최고인 것들을 이용해 사전 학습을 진행하여 BERT의 성능을 뛰어 넘었다. MASS는 마이크로소프트에서 만든 모델로 역시 BERT와 다른 사전 학습 방식을 통해 더 뛰어난 성능을 보여준다. BERT의 경우 Seq2Seq에서 사용된 인코더(Encoder)를 이용해 언어를 이해하도록 미리 학습하고, GPT는 디코더(Decoder)를 이용해 사전 학습을 진행하는 반면, MASS는 인코더와 디코더, 그리고 어텐션 매커니즘까지 활용해 사전 학습을 진행한다. 인코더의 입력 중 원하는 만큼을 생략하고 어텐션 레이어를 거쳐서 디코더에 상태 값을 전달한 뒤, 생략된 입력 값을 디코더에서 추측하도록 훈련시키는 방식이다. 이 때, 인코더에서 입력을 1개만 가리는 경우 BERT의 사전 학습 모델을 모방할 수 있고, 모든 입력을 생략하면 GPT의 사전 학습을 모방할 수 있다.

3. 연구 방법

딥러닝을 이용하여 실제 사람처럼 뉴스에 반응하는 텍스트를 생성하는 시스템을 구현하려면, 고정된 말뭉치(corpus)가 아닌, 시간에 따라 변하는 뉴스와 그의 댓글로 이루어진 학습 세트(training set)가 있어야 한다. 이러한 학습 세트의 구성을 위해, 매일 다음(Daum)포털의 뉴스 섹션에서 댓글 많은 뉴스 상위 50개의 뉴스와 그에 달린 모든 댓글과 대댓글(댓글의 댓글)을 크롤링하여 학습 세트로 저장하였다. 2018년 2월 1일 부터 2019년 8월 29일 까지 약 28000개의 뉴스에 대한 데이터를 수집했다.

우리의 이전 연구에서는 이렇게 수집한 뉴스와 댓글 데이터를 띄어쓰기 단위로 워드 임베딩한 후 트레이닝에 사용하였다. 한글은 단어에 조사가 붙고 띄어쓰기를 하므로, 띄어쓰기 단위로 워드 임베딩 할 경우, 같은 단어라도 다른 조사가 붙은 경우는 서로 다른 단어로 간주된다. 이로 인해 일정량의 문장 내에 존재하는 '단어'의 숫자가 상당히 커지게 되고, 그에 따른 메모리의 사용량도 늘어나게 되어, 제한된 적은 양의 데이터를 이용하여 학습할 수밖에 없었다. 그 때문에 누적된 데이터가 많아도

이를 모두 활용하는데 어려움이 있었다.

또한 NMT 모델이 LSTM을 기반으로 구성 되어 있고, RNN/LSTM은 태생적으로 학습에서의 병렬성(parallelism)이 비교적 낮기에, 학습시의 GPU의 활용률(utilization) 역시 비교적 낮은 수치를 보이게 된다. 그 때문에, 매일 수집한 수백 개의 뉴스와 수만 개에서 수십만 개의 댓글 데이터를 학습하여 당일의 뉴스에 대한 댓글을 실시간으로 생성해야 하는 환경에서는, NMT의 느린 학습 속도가 병목 지점이 되었다.

3.1 형태소 분석

띄어쓰기 단위가 아닌 형태소 단위로 단어를 구분 할 경우, 주어진 문장에 존재하는 고유한 단어의 수가 대폭 줄어드는 효과가 있다. 이는 더 큰 학습 세트를 대상으로 학습을 진행할 수 있게 해주므로, 더 자연스러운 댓글을 생성할 수 있게 한다.

형태소 분석기를 선택하기 위해 2018년 카카오에서 개발한 Khaii와[23], 일본 교토대학 정보학 연구과와 일본 NTT 커뮤니케이션 과학 기초 연구소가 공동으로 개발하고 한국의 '은전한닢' 프로젝트에서 한국어용으로 적용시켜 널리 사용되고 있는 MeCab을[24] 고려하였다. Khaii의 경우 MeCab에 비해 더 작은 형태소 부분까지 분석하는 경향이 있기에, 전체의 고유 단어 숫자를 더 줄여주는 효과가 있지만, 분석된 형태소의 의미가 중복되는 경우가 많아질 수 있기에, 워드 임베딩의 의미가 모호해 질 수 있다고 판단해 MeCab을 선택하였다.

[Table 1] A comparison of vocabulary sizes with news articles for 120 days, by whether using MeCab

	Size of Vocabulary
Not Using MeCab	2,771,643
Using Mecab	134,422

사용결과 120일 분량의 뉴스와 댓글에 대해 띄어쓰기 단위로 단어를 구분할 경우 2,771,643개의 어휘수가 나오는 반면 MeCab을 이용한 형태소 분석 결과로 단어를 구분할 경우는 134,422개 정도의 어휘수가 나왔다. 또한 형태소 분석기를 이용하면 맞춤법이 틀린 문장에 대해서는 글자 단위로 쪼개지기 때문에 학습 데이터양이 많아지더라도 총 어휘 개수는 조금씩 상승했다. 때문에 많은 양의 학습 데이터를 이용했을 때 생기는 어휘 수 증가 문제점이 상당히 해결되었다.

3.2 학습 초기값 설정

본 연구의 시스템은 매일 최신 기사를 보고 댓글을 생성해내야 하기 때문에 모델 또한 매일 새롭게 학습해야 한다. 그러나 학습할 수 있는 시간은 한정되어 있으므로 지금까지 수집한 모든 기사와 댓글을 학습 데이터로 사용할 수는 없다. 그래서 우리는 가장 최근 수집된 수백 개 기사의 데이터를 학습에 사용하기로 했다. 하루가 지날 때마다 새로운 기사들이 학습 데이터에 추가되고, 기존 학습 데이터에서 가장 옛날 기사를 제외시키기 때문에 학습 범위가 슬라이딩 윈도(sliding window)의 형태를 가진다고 볼 수 있다. 이 경우 바로 전날에 모델이 학습한 가중치를 초기 값으로 설정하면 학습시간을 더 빠르게 학습할 수 있었다. 전날 학습된 모델로 일종의 전이학습(Transfer Learning)을 하는 셈이다.

3.3 트랜스포머 모델

더 나은 학습 속도와 댓글 생성 결과를 위해 본 연구에서는 구글에서 구현한 트랜스포머 모델을 사용했다[25]. 구글에서 제공하는 기본 파라미터를 사용했고 에폭(epoch) 수는 5로 설정하였다. 먼저 데이터에 형태소 분석을 적용하지 않고 띄어쓰기 단위로 단어를 구분하여 학습을 시킨 결과, 학습 속도는 굉장히 빨랐으나 어휘 숫자가 200만개에 육박할 정도로 커지고, 그에 따른 메모리 사용량의 증가 때문에 4일치 데이터 정도만을 학습에 사용할 수 있었다. 형태소 분석을 적용한 뒤에 학습을

시킨 경우에는 120일치가 넘는 뉴스 데이터를 학습에 사용할 수 있었다. 하지만 새벽동안 데이터를 수집하고 모델을 학습한 후 아침부터는 실시간으로 뉴스에 대한 댓글을 생성해야하기 때문에 학습에 할애할 수 있는 시간이 6시간 정도로 한정됐다. 때문에 정해진 시간 내에 학습을 완료하도록 학습 데이터양을 120일치 뉴스에 대한 내용과 댓글로 제한했다.

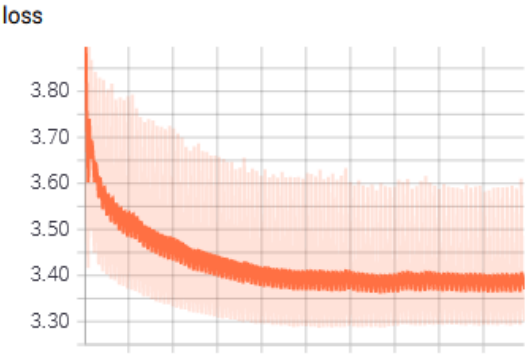
3.4 에러 허용 시스템

사회적으로 여러 이벤트와 이슈가 발생함에 따라, 뉴스와 댓글 데이터 수집시 뉴스 제공 포탈 페이지의 레이아웃이 바뀌는 경우가 종종 발생했다. 이 때문에 데이터 수집 프로그램 또는 실시간 뉴스를 읽는 프로그램이 새로운 레이아웃을 인식하지 못해 오작동을 하는 경우가 있었다. 이러한 프로그램들의 오작동은 모델을 학습 시키거나 실시간으로 포털 시작 페이지의 뉴스에 대한 댓글을 생성하는 다음 단계들에 영향을 미쳐 전체 자동화 시스템의 동작을 멈추고 오류를 발생했다. 이를 해결하기 위해 자동화 프로그램을 수정해 어떤 단계에서 오류가 난 경우더라도 진행을 멈추지 않고 전날 학습시켰던 모델을 이용하거나 오류 난 단계를 생략해 다음 단계에 영향을 미치지 않고 상시 운영되도록 구현하였다.

4. 연구 결과

본 연구에서는 Python 3.5.2 언어를 사용해 시스템을 구성했으며, 머신러닝 프레임워크로는 Tensorflow 1.13.0-dev20190220 을 사용했다. 이 시스템은 Intel Core i7-7700 3.60GHz, 64GB RAM, NVIDIA GTX 1080ti 2개로 이루어진 하드웨어 환경에서 작동했다. 매일 0시 1분에 시스템을 실행해서 전날 다음 포털에서 가장 댓글이 많았던 기사들 상위 50개를 수집하고, 수집이 완료되면 수집한 기사들을 변형해 학습 데이터들을 만들어 낸 후, NMT 모델과 트랜스포머 모델을 동시에 학습한다. 두 모델 모두 학습이 완료되면 작성 스케쥴을 생성해 스케쥴러에 등록한다. 이후 스케쥴러에 등록된 시간이 오면 현재 다음 포털 메인 페이지에 있는 기사 제목중 하나를 추출하고, 학습된 두 모델들을 사용해 그 제목에 대한 반응을 생성해서 기사 링크와 함께 트위터에 게시한다.

4.1 학습 그래프



[Fig. 4] A training loss graph of Transfomer model

새롭게 추가한 트랜스포머 모델의 학습 그래프이다. 모델은 6개의 레이어와 8개의 멀티 헤드로 이루어져 있고 2.0의 학습률로 시작해서 약 6시간 동안 학습했다. 학습 초기에는 오차 값이 계속 감소하지만 4번째 반복부터는 학습을 진행해도 오차값에 큰 변화가 생기지 않았다. 어떤 기사에 대한 사람들의 반응들엔 공통적인 기준이 없이 다양할 수 있기 때문에 모델이 더 이상 학습하지 못한 것으로 보인다. 학습 도중 짧은 주기로 오차 값이 계속 줄었다 상승했다 하는 이유는 같은 뉴스에 대한 댓글들을 학습 할 때는 오차가 하락하지만 다른 뉴스로 넘어가서 다른 뉴스의 댓글들을 학습할 때는 오차가 다시 상승하기 때문으로 추정된다.

4.2 이전 모델과 비교

시스템에 트랜스포머 모델을 추가한 결과 기존의 NMT 모델에 비해 더 자연스러운 반응을 생성할 수 있었다.

[Table 2] A comparison of generated comments of both models

Title	제주 다금바리 어획량 5년새 10배↑..횟집가격 20만원대 왜?
NMT	이 나라 는 절대 안 된다 . . . . .
Transformer	제주 도 안 가 고 싶 다

[Table 3] A comparison of generated comments of both models

Title	한국당에 막힌 검경수사권 조정안..국회 접수여부 해석 분분
NMT	니 들 이 잘 하 는 게 아니 라 국민 들 이 심판 하 는 것 이 다 . . 국민 들 이 심판 하 는 게 아니 라 .
Transformer	토착 왜구 들

[Table 4] A comparison of generated comments of both models

Title	"선생님 볼에 뽀뽀해볼래" 女제자 4년간 성폭행..징역 9년 확정
NMT	여자 가 아니 라 . .
Transformer	우리 나라 는 법 이 너무 물러

기존의 NMT 모델이 생성한 반응에는 반복되는 어구나 말줄임표가 많이 포함되는 반면 트랜스포머 모델의 결과는 상대적으로 완전한 문장을 이루는 경우가 많았다.

4.3 형태소 분석 결과

[Table 5] A morphological analysis example

Before analyze	조은누리양의 기적 생환이 지금 국민에게 엄청난 힘을 주고 있다.
After analyze	조 은 누리 양 의 기적 생환 이 지금 국민 에게 엄청난 힘 을 주 고 있 다

[Table 6] A morphological analysis example

Before analyze	온국민의 관심과염원으로 기적이 일어났읍니다.!! 이젠 일본제품 불매운동으로 또 다른 기적을 만들어갈때입니다.!!
After analyze	온 국민 의 관심 과 염원 으로 기적 이 일어났 읍니다 . !! 이젠 일본 제품 불매 운동 으로 또 다른 기적 을 만들 어 갈 때 입니다 . !!

이번 연구에서는 형태소 분석기로 MeCab을 사용했다. 이를 통해 띄어쓰기 단위로 입력을 처리하던 방식을 형태소 단위로 처리하고 문법에 맞지 않는 단어는 글자 단위로 취급했다. 이를 통해 처리하는 단어의 수가 상당이 감소됐고 학습 할 수 있는 데이터가 많아 져 모델의 성능을 높일 수 있었다. [Table 5]를 보면 명사 뒤에 붙어 의미를 변화시키는 조사들이 별도의 어휘로 분리되는 것을 볼 수 있다. 또한 [Table 6]에서는 띄어쓰기가 적절히 되어 있지 않은 문장을 문법에 맞게 띄어쓰기를 넣어 처리하는 것을 볼 수 있다.

4.4 트위터 반응

지난 2년 동안 트위터에 뉴스링크와 함께 생성한 댓글을 올리는 자동화된 에이전트를 운영하며 860여개의 트윗을 올렸다.

BLEU[26], METEOR[27] 등의 객관적인 평가 방법이 존재하는 기계 번역 분야와 달리, 반응 생성 분야에서는 객관적으로 쓰이는 평가 방법을 찾기가 쉽지 않다. 또한 기존 기계 번역 분야의 평가 방법이 반응 생성 모델을 평가하기엔 적합하지 않다는 것을 보이는 연구도 존재한다.[28] 때문에 우리는 기존의 방법을 대신해 트위터에 생성한 댓글들을 공개하여 사용자들의 반응을 확인했다. 매일 실시간 뉴스를 대상으로 댓글을 생성한 결과 총 3개의 좋아요, 7개의 리트윗, 2명의 팔로워가 생겼다.

[Fig. 5] Reactions to uploaded tweets on Twitter

5. 결 론

LSTM 기반 NMT 모델기준으로 어휘수의 과다 때문에 3일치 뉴스와 댓글 데이터 밖에 학습하지 못한 문제를 형태소 분석을 통해 100일 이상의 데이터를 한 번에 학습할 수 있게 되었다.

그러나 매일 실시간 뉴스에 대응하는 댓글을 생성하기 위해 정해진 시간 안에 학습을 완료해야 했다. 때문에 학습의 병렬화에 한계가 있는 RNN/LSTM 모델에서는 50일 정도의 데이터로 학습량을 제한해야했다. 이에 반해 새로 시스템에 적용한 트랜스포머 모델은 RNN이 아닌 FNN(Fully Connected Network) 기반으로 셀프-어텐션 레이어가 구성되어있어 학습의 병렬성이 뛰어나다. 따라서 더 짧은 시간 안에 120일치 데이터로 학습이 가능했다.

또한 트랜스포머 모델이 그 이전의 LSTM 기반 모델보다 주어진 입력 문장에 대하여 더 연관된 결과를 생성했다. 뿐만 아니라 동일 시간 대비 더 많은 데이터에 대해 학습할 수 있기 때문에 단기간 집중된 이슈가 아닌 빈도가 적은 사건들에 대해서도 관련된 댓글을 생성했다.

현재 자연어 처리 분야는 성능이 개선되고 새로운 개념이 적용된 모델이 끊임없이 등장하고 있다. 때문에 현재 시스템에 이러한 새로운 모델과 개념들을 적용한다면 지금보다도 사람에 가까운 댓글들을 생성할 수 있을 것이라 예상된다.

이번 연구에서도 댓글을 포탈에 직접 작성하지 못하기 때문에 다른 사용자들의 반응을 보는데 한계가 있었다. 현재 결과를 업로드 하는 데만 트위터를 사용하고 있는데, 향후엔 다른 사용자들을 팔로우하고 다른 트윗을 리트윗하는 과정 또한 학습해 이용자들과 소통을 할 예정이다. 이를 통해 생성한 텍스트에 대한 다른 이용자들의 반응을 이전보다 많이 수집할 수 있을 것으로 예상된다.

ACKNOWLEDGMENTS

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

REFERENCES

[1] J. Choi, S. Sung, K. Kim. “A Study on Automatic Comment Generation Using Deep Learning”, Journal of Korea Game Society, 18(5), pp 83-92, 2018.
[2] Stroh, Eylon, and Priyank Mathur. "Question answering using deep learning.", 2016
[3] Tang, Gongbo, et al. "Why self-attention? a targeted evaluation of neural machine translation architectures.", arXiv preprint arXiv:1808.08946, 2018.
[4] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate.", arXiv preprint arXiv:1409.0473, 2014.
[5] Vaswani, Ashish, et al. "Attention is all you need.", Advances in neural information processing systems, pp.5998-6008, 2017.
[6] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding.", arXiv preprint arXiv:1810.04805, 2018.
[7] Radford, Alec, et al. "Improving language understanding by generative pre-training.", https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf, 2018.
[8] Yang, Zhilin, et al. "XLNet: Generalized Autoregressive Pretraining for Language Understanding.", arXiv preprint arXiv:1906.08237, 2019.
[9] Song, Kaitao, et al. "Mass: Masked sequence to sequence pre-training for language generation.", arXiv preprint arXiv:1905.02450, 2019.
[10] Sherstinsky, Alex. "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network.", arXiv preprint arXiv:1808.03314, 2018.
[11] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory.", Neural computation 9.8, pp.1735-1780, 1997.
[12] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv: 1412.3555, 2014.
[13] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks.", Advances, in neural information processing systems, 2014.
[14] Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional recurrent neural networks.", IEEE Transactions on Signal Processing 45.11, pp.2673-2681, 1997.
[15] Cheng, Jianpeng, Li Dong, and Mirella Lapata. "Long short-term memory-networks for machine reading.", arXiv preprint arXiv:1601.06733, 2016.
[16] Jakob Uszkoreit, “Transformer: A Novel Neural Network Architecture for Language Understanding”, https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html, 2017
[17] Damien Sileo, “Understanding BERT Transformer: Attention isn’t all you need”, https://medium.com/synapse-dev/understanding-bert-transformer-attention-isnt-all-you-need-5839ebd396db, 2019.
[18] https://reniew.github.io/43/
[19] leod. “Generate Hacker News Comments from Titles”, https://github.com/leod/hncynic.
[20] Shibata, Yusuxke, et al. “Byte Pair encoding: A text compression scheme that accelerates pattern matching.”, Technical Report DOI-TR-161, Department of Informatics, Kyushu University, 1999.
[21] “The Stanford Question Answering Dataset”, https://rajpurkar.github.io/SQuAD-explorer/
[22] Peters, Matthew E., et al. “Deep contextualized word representations.", arXiv preprint arXiv: 1802.05365, 2018.
[23] kakao, “Kakao Hangul Analyzer III”, https://github.com/kakao/khaiii
[24] eunjeon,“mecab-ko-dic”, https://bitbucket.org/eunjeon/mecab-ko-dic/src/master/
[25] tensorflow, “Models and Examples built with Tensorflow”, https://github.com/tensorflow/

models
[26] Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation.", Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002.
[27] Banerjee, Satanjeev, Alon 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. 2005.
[28] Liu, Chia-Wei, et al. "How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation.", arXiv preprint arXiv: 603.08023, 2016.

성 소 윤 (Seong, So-yun)

약 력 : 2019 한국산업기술대학교 게임공학과 학사
2019-현재 한국산업기술대학교
디지털엔터테인먼트학과 석사과정

관심분야 : 게임 프로그래밍, AI

최 재 용 (Choi, Jae-yong)

약 력 : 2019 한국산업기술대학교 게임공학과 학사
2019-현재 한국산업기술대학교
디지털엔터테인먼트학과 석사과정

관심분야 : 기계학습, 게임 프로그래밍

김 경 철 (Kim, Kyoung-chul)

약 력 : 1992 KAIST 과학기술대학 전산학과 학사
1994 KAIST 정보및통신공학과 석사
2005 KAIST 전산학과 박사
2001-2005 ㈜고누소프트 가약스부문 차장
2006-현재 한국산업기술대학교 게임공학부 부교수

관심분야 : 컴퓨터구조, 분산처리, 온라인 게임 서버