BERT을 이용한 한국어 문장의 스타일 변화

이주성, 오연택, 변현진, 민경구

LG사이언스파크

m3155@naver.com, yeontaek.oh@lgsp.co.kr, hyunjin.byun@lgsp.co.kr, kyungkoo.min@lgsp.co.kr

Controlled Korean Style Transfer using BERT

Joosung Lee, Yeontaek Oh, Byun hyunjin, Kyungkoo Min

LG Sciencepark

요 약

생성 모델은 최근 단순히 기존 데이터를 증강 시키는 것이 아니라 원하는 속성을 가지도록 스타일을 변화시키는 연구가 활발히 진행되고 있다. 스타일 변화 연구에서 필요한 병렬 데이터 세트는 구축하는데 많은 비용이 들기 때문에 비병렬 데이터를 이용하는 연구가 주를 이루고 있다. 이러한 방법론으로 이미지 분야에서 대표적으로 cycleGAN[1]이 있으며 최근 자연어 처리 분야에서도 많은 연구가 진행되고 있다. 많은 논문들이 사용하는 데이터도메인은 긍정 문장과 부정 문장 사이를 변화시키는 것이다. 본 연구에서는 한국어 영화리뷰 데이터 세트인 NSMC[2]를 이용한 감성 변화를 하는 문장생성에 대한 연구로 자연어 처리에서 좋은 성능을 보여주는 BERT[8]를 생성모델에 이용하였다.

주제어: 문장 생성, 스타일 변화, 감성 분석

1. 서론

데이터 생성은 데이터 증강 관점에서 많이 연구되고 있고 증강된 데이터를 이용하여 딥러닝 학습에 활용하기도 한다. 하지만 데이터 증강은 이미지 생성에서 연구가 많이 되어 왔고 언어 생성은 상대적으로 연구가 적은 상황이다.

자언어 생성은 이미지 생성과 다르게 불연속한 영역에서 이뤄지게 된다. 따라서 이미지 영역에서 쓰이는 GAN 기법을 그대로 적용해서는 학습이 제대로 이뤄지지 않는 어려운 점이 있다[3]. 따라서 정책 경사값(Policy Gradient)을 이용하거나 분류기(Discriminator)를 이용한 적대적 학습(Adversarial Training) 방법 등이 존재한다[3]. 본 논문은 생성 문장의 토큰의 소프트 단어 임베딩(Soft Word Embedding)을 감성 분류기에 적용시켜서 얻은 결과로 End-to-End 학습을 진행하여 불연속 공간의 학습 문제를 해결하고자 한다.

최근 문장 생성 연구에서 문장 스타일 변화 연구들이 활발히 진행되고 있다. 스타일 변화를 시도할 때, 병렬 데이터가 많으면 Seq2Seq[11]를 통하여 해결할 수 있다. 하지만 실제로 병렬 데이터를 모으는 것은 쉽지 않기에 많은 최근 연구들은 비병렬 데이터세트로 스타일(style) 변화를 시도한다[4, 5, 6, 7, 9, 13]. 비병렬 데이터세트를 이용한 연구들은 스타일에 의존적이지 않은 문장의 잠재 내용 표현(Latent Content Representation)이 같도록 하는 방법들이 주를 이룬다[4, 5, 6]. 이 방법들은 내용 표현(Content Representation)에 적대적 학습을 적용하여 효과를 얻을 수 있다. 하지만 [7]에서는 문장들의 내용 보존을 위하여 적대적 학습을 적용하여도 내용 표현이 스타일에 독립적이지 않음을 보여준다. 따라서 본 논문에서는 적대적 학습에 초점을 맞추지 않고 재변환(Back-Translation)을 이용한 손실(loss)을 이용하여 적대적 학습으로 얻을 수 있는 효과를 간접적으로 추가하였다.

감성 분석을 하는 분류기는 NSMC(Naver Sentiment Movie Corpus)[2] 데이터를 이용하여 학습하였다. 분류기 모델은 CNN 모델을 이용하였으며 생성모델 학습에 관여하는 CNN 모델인 분류기-G와(Discriminator-G) 생성 문장 평가만을 위해 같은 구조를 가지는 분류기-B(Discriminator-B), 두 가지를 구축하였다. 실험 결과 표 3으로 생성된 문장들은 분류기-G에 적합이 되어서 감성을 가지는 문장이 생성됨을 알 수 있다. 따라서 생성된 문장이 올바른 감성을 가지는지는 분류기-B를 이용하여 추가적인 평가를 진행하였다.

본 논문에서는 양방향(Bi-directional) 언어모델(Language modeling)인 BERT을 이용하여 문장을 인코딩을 한다. 생성 모델은 단방향 모델이기 때문에 일반적으로 문맥적 임베딩(contextual embedding)을 사용할 수 없다. 그러나 스타일 변화 문장 생성 작업의 인코딩 과정에서는 입력문장 전체가 주어지므로 양방향 모델인 BERT를 적용할 수 있다.

2. 관련 연구

BERT는 최근 다양한 자연어 처리 분야에서 SOTA(state-of-the-art)를 달성하고 있는 언어모델이다. BERT는 트랜스포머(Transformer)[10]로 구성된 모델로 RNN과 달리 입력시퀀스가 병렬적으로 처리되는 방식이다. BERT는 많은 양의 말뭉치를 이용하여 모델을 학습하여 다운스트림 작업에서 큰 효과를 보여준다. 구글에서는 영어뿐 아니라 다양한 언어를 지원하는 다국어 버전 BERT도 제공하고 있다. 다국어 버전의 BERT는 한국어

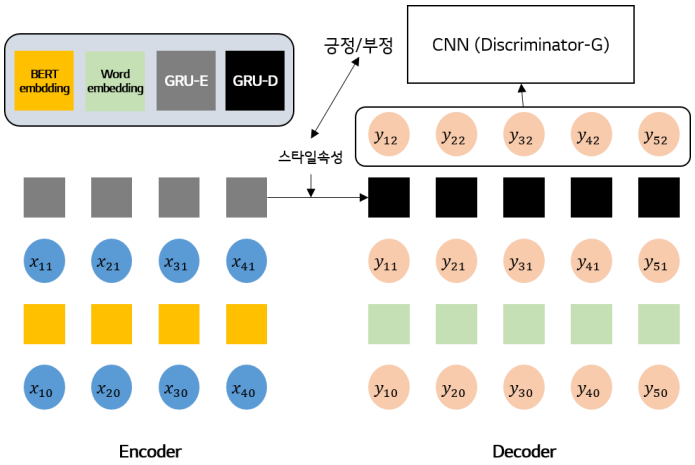

그림 1. 모델의 구성 (Encoder, Decoder, Discriminator)

기계 독해 작업 KorQuAD[14]에서도 좋은 성능을 보인다[15].

문장 생성의 대표적인 연구는 SeqGAN이 있다[3]. 이 논문에서는 GAN학습시 언어공간의 불연속성 문제를, 정책 경사값를 이용하여 해결 하였다.

NSMC을 이용한 감성 문장 생성으로 [9]의 연구가 있다. 이 연구는 [4]을 기반으로한 VAE 모델을 이용하여 주어진 문장을 속성 z(Context)와 속성 c(Style)로 나누어서 문장을 표현하였다. 문장의 스타일은 다르지만 내용은 동일한 표현을 가지도록 적대적 학습을 하였다.

3. 모델 및 학습 과정

3.1 모델 구성

본 연구에서 제안한 모델은 그림 1과 같이 인코더(Encoder)와 디코더(Decoder), 분류기(Discriminator)로 구성된다. 모델은 전체적으로 깊은 구조가 아닌 한 개 층의 GRU로 구성된다.

인코더는 사전 학습된 다국어-BERT을 이용한 문맥적 임베딩과 GUR-E(Encoder)로 구성되어있다. GRU-E 토큰의 은닉 상태 벡터는 GRU-D(Decoder)의 초기값에 사용된다.

디코더는 생성모델로 단방향 모델이기 때문에 뒷 부분의 토큰을 모르는 상태에서 문맥적 임베딩을 사용할 수 없다. 따라서 디코더에서는 단어 임베딩 행렬(Word Embedding Matrix)을 사용해야 한다.

분류기의 입력은 디코더에서 사용된 단어 임베딩을 거쳐 CNN 구조인 감성 분류기 학습을 한다. 분류기는 분류기-G와 분류기-B가 존재하는데 두 개의 모델은 CNN을 기반으로 같은 구조를 가지나 토큰을 BERT 임베딩을 사용하는지, 단어 임베딩 행렬을 사용하는지의 차이이다.

모델을 구성하는 매개변수(parameter)들은 표 1과 같이 구성된다. BERT는 구글에서 제공한 Multilingual-Cased을 사용하였고 토큰은 768의 차원으로 임베딩이 된다. 단어 임베딩 행렬, GRU-E, GRU-D 또한 768의 차원을 가지도록 설정하였다. 분류기는 CNN 커널 사이즈를 2부터 5까지 각각 100 특징맵(Feature Maps)을 가지도록 설정하였다.

	차원	Kernel
Multilingual-BERT cased	768	x
Word embedding	768	x
GRU-E	768	x
GRU-D	768	x
CNN	100 (feature maps)	2~5

표 1. 모델을 구성하는 모듈의 차원

모델에서 디코더 부분은 Teacher-forcing과 No Teacher-forcing 두 가지 방법으로 진행을 할 수 있고 각각 디코더와 생성자에서 사용이 된다. 즉 Teacher-forcing은 문장을 재구성 할 때 사용을 하게 되고 재변환 혹은 모델의 추론과 같은 과정에서는 No Teacher-forcing 방식으로 진행하게 된다.

3.2 학습 과정

학습 과정은 그림 2, 3와 같으며 GAN과 유사하게 모듈을 번갈아 학습을 진행하며 총 3개의 단계로 구성된다.. 손실은 재구성 손실(reconstruction loss), 재변환 손실(back-translation loss), 분류기 손실(discriminator loss)이 존재하고 각 단계에서 학습하는 매개변수들은 다르다.

그림 2는 학습의 step-1을 의미하고 인코더와 디코더의 매개변수들을 학습하게 된다. 학습에 사용되는 손실은 재구성 손실과 재변환 손실의 가중치 합을 최소화 하도록 진행하며 식 (1)과 같이 구성이 된다. 가중치 w는 0의 값으로 시작하여 학습 진행에 따라 점점 커지도록 한다. 따라서 모델의 학습 초기에는 재구성에 가중치를 주고 학습이 진행 될수록 재변환에 가중치를 주도록 하였다.

$$\text{Step1: loss} = (1-\text{w}) * \text{recon}_{\text{loss}} + (1+\text{w}) * \text{bt}_{\text{loss}} \quad (1)$$

재구성 손실은 입력 문장 x_1이 인코더와 디코더를 거쳐 복원이 되도록 하는 손실함수로 식 (2)을 따른다. 입력 문장이 GRU-E으로 들어가 마지막 은닉 상태 벡터를 도출하고 이 결과를 인코딩 벡터(encoding vector)라 한다. 디코더에 x_1이 입력으로 다시 들어가서 복원 과정을 거칠 때 인코딩 벡터와 입력 문장 스타일 c_1을 결합한 값이 GRU-D의 초기값이 된다. Step-1을 통하여 문장 토큰들의 조건부 확률과 단어 임베딩 행렬을 학습하는 효과가 있다.

$$\text{recon}_{\text{loss}} = Cross\_entropy(x_1, Dec(Enc(x_1), c_1)) \quad (2)$$

재변환 손실은 먼저 입력 문장 x_1의 인코딩 벡터와 입력 문장의 반대되는 스타일 c_2를 결합하여 생성자로 문장 y_2를 생성한다. 생성 문장 y_2를 다시 인코딩후 기존 입력 문장 스타일 c_1을 결합하여 문장 y_1을 생성한다. y_1과 x_1이 같은 문장이 되도록 손실함수를 설정하는 것이 재변환 손실이 되며 식 (3)과 같다. 재변환은 기계 번역

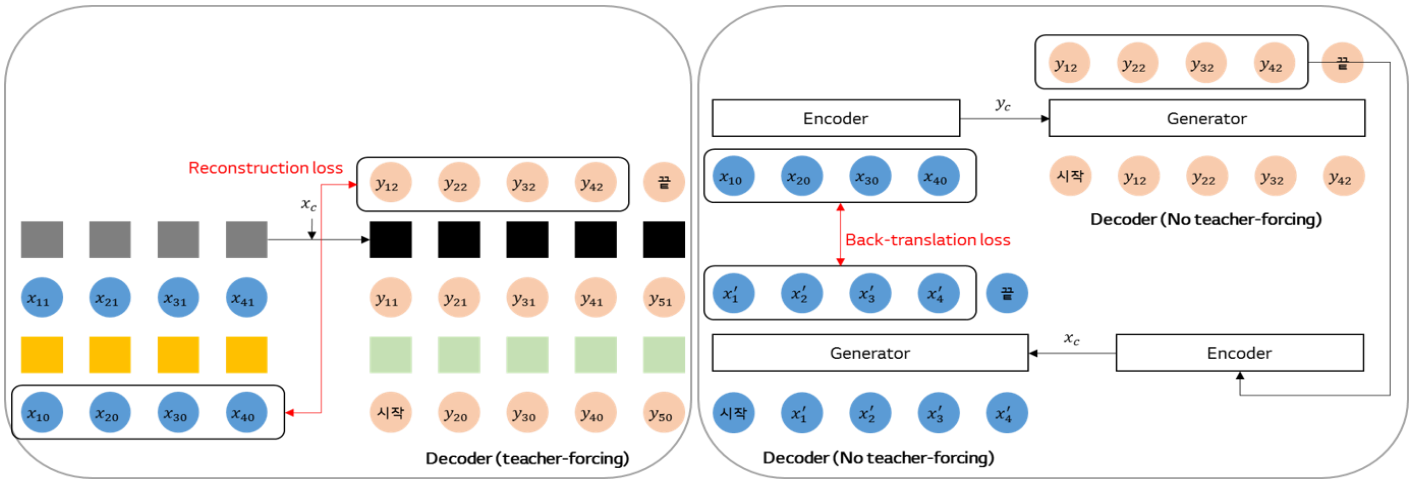

그림 2. 모델 학습 step-1(Reconstruction loss, Back-translation loss)

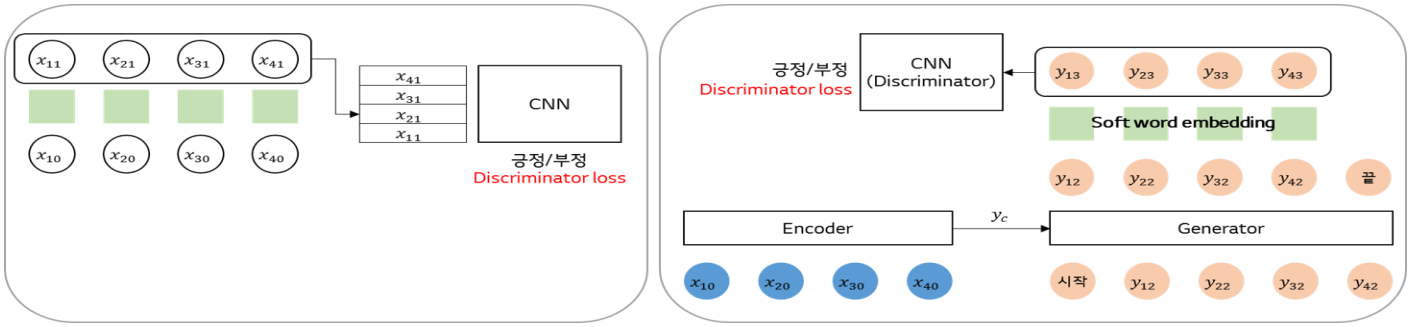

그림 3. 모델 학습 step-2(Original sentence Discriminator loss), step-3(Generated sentence Discriminator loss)

$$\mathrm{bt}_{\mathrm{loss}} = Cross\_entropy(x_1, Dec(Enc(Dec(Enc(x_1), c_2)), c_1)) \quad (3)$$

문제에서 비지도 학습 방법으로 사용되는 방법이다. 즉 재변환 손실은 입력 문장과 스타일이 다른 문장 도메인으로 변환되었다가 입력 문장 도메인으 f 로 다시 돌아와야 하므로 인코딩이 기존 스타일 및 다른 스타일 모두를 생성할 수 있게 학습되는 효과가 있다. 따라서 인코딩 벡터 공간은 스타일 속성에 덜 영향을 받게 된다[13].

Step-2는 그림 3에서 왼쪽에 해당하는 과정으로 분류기를 학습하는 것으로 식 (4)와 같다. 학습되는 매개변수들은 CNN의 커널 매개변수들과 CNN 출력 벡터를 사전(vocab) 차원으로 변경시켜주는 행렬이다. Step-2에서는 단어 임베딩 행렬은 학습하지 않는다. 분류기를 학습하는 것은 step-3에서 생성 문장이 제대로 된 스타일을 가지도록 도움을 주기 위한 것으로 분류기의 성능을 위해 단어 임베딩이 적합되도록 학습시키지 않았다.

$$\text{Step2: Discriminator}_{\mathrm{loss}} = Cross\_entropy(x, c) \quad (4)$$

Step-3에서는 step-2에서 학습된 분류기를 이용하여 생성모델을 이어서 학습하는 것으로 그림 3의 오른쪽, 식 (5)와 같다. 학습되는 매개변수들은 단어 임베딩 행렬을 제외한 인코더와 디코더 매개변수들이다. 입력 문장 $\mathbf{x}_1$에 대해 step-2에서 생성 문장을 만드는 것까지는 같은 과정을 거친다. 그 뒤로 생성 문장에 대한 분류기 손실을 계산한다. 계산된 분류기 손실에서부터 end-to-end로 인코더와 디코더를 학습하게 되며 역전파가 되기 위해서는 CNN 네트워크부터 인코더까지 과정이 미분이 가능해야 한다. 따라서 생성자에서 생성된 출력 벡터 확률을 이용한 소프트 단어 임베딩을 진행하였다. 이 과정을 통하여 언어 생성의 불연속 공간 문제를 해결하는데 도움을 준다.

$$\text{Step3: Discriminator}_{\mathrm{loss}} = Cross\_entropy(Dec(Enc(x_1), c_2), c_2) \quad (5)$$

학습은 3 epoch 동안 진행하였으며 Batch Size는 32로 설정하였다. Learning rate의 초기값은 0.001으로 설정하여 5,000 Iteration당 반으로 감소시켰다. Optimization은 Adamax[12]을 사용하였다.

4. 실험 및 결과

실험 데이터는 NSMC로 영화 리뷰 감성 데이터 실험용 학습 데이터는 150,000 문장이 주어지며 테스트 데이터는 50,000 문장이 주어진다. 표 2는 데이터 샘플들이며 문법과 띄어쓰기가 잘되어 있는 데이터들도 존재하고 그 반대인 경우도(영어 포함 등) 존재하는 데이터 세트임을 보여준다.

감성 분류모델은 생성모델 학습 과정에서 학습한 분류기-G(Generator)와 생성모델과 관계없이 학습한 분류기-B(BERT) 두 개가 있다. 분류기-B는 분류기-G와 똑같은 CNN 구조이나 토큰을 벡터화 하는 과정을 BERT 임베딩을 이용하였다.

성능 평과 결과는 표 3과 같다. 분류기-B 또한 생성모델 학습할 때와 같이 3 epoch을 학습하였다. 테스트 데이터세트에서는 분류기-G 보다 분류기-B가 성능이 좀 더 좋았다. 분류기-B는 BERT 모델을 사용하였기 때문에 문맥적 임베딩을 사용하였고 분류기-G은 단어 임베딩을 따로 학습하지 않은 결과로 해석된다. 반면 생성 문장에서는 분류기-G가 더 좋은 결과를 도출한다.

이는 학습과정의 step-3에서 인코더와 디코더 부분이 분류기-G를 손실 값으로 사용하였기 때문이다. 즉 생성문장이 분류기-G 기준에서 맞는 감성을 가지도록 영향을 받아 생성되었음을 확인할 수 있다.

표 3에서 평가에 사용된 생성 문장 98,684개는 NSMC 테스트 데이터 문장에 대해서 스타일을 같게 주었을 때와 다르게 주었을 때 두 가지 경우 모두에 대해 생성된 문장들이다. 학습 데이터로 추가적인 데이터를 생성하는 것보다 학습 때 아예 보지 못한 테스트 데이터를 다루는 것이 일반적이다. 따라서 생성 문장으로 분류기 성능을 측정할 때 학습 데이터는 이용하지 않았다. 테스트 데이터에 있는 모든 문장에 대해 생성을 한다면 100,000개가 되어야 하지만 1,316문장은 BERT 토크나이저로 토큰화를 하는 과정에서 문제가 있어 제외하였다. 문장의 띄어쓰기가 아예 안되어 있거나 사전에 없는 캐릭터의 나열들은 [UNK] 토큰으로 처리가 되기 때문이다[표 4]. 최대 생성 길이는 20으로 제한하여 문장을 생성하였다.

표 5, 6은 생성 문장들의 샘플들이다. 샘플들 외에도 입력 문장과 생성문장들은 같은 토큰으로 시작하는 경우가 대부분이며 비슷한 내용을 담는 것을 관찰할 수 있다.

	부정	긍정
일반 데이터	주연배우들의 이름만 기억에 남.	상당히 재밌게 봤습니다
비정제 데이터	ㄴㅈ5점대 asfgsdlgkbjsjvb	아름다운 Andrew McCarthy^^*

표 2. NSMC 데이터 세트

	Test set 평가 (50000개)	생성 문장 평가 (98684개)
Discriminator-G	83.95%	**85.74%**
Discriminator-B	**84.25%**	85.22%

표 3. 감성 분류 성능

입력문장	토근화
괜찮네요오랜만 포켓몬스터잼있어요	[UNK]
정말 재밌게 잘봤습니다	정', '##말', '[UNK]', '잘', '##봤', '##습', '##니다', '!'

표 4. 테스트 데이터의 문장 토큰화

입력문장 (긍정)	영화속 상황과 지금 내 현실이 너무나 비슷해 질질 짜버렸네;;
생성문장 (긍정)	영화중 하나하나가 너무 아름다운 영상미와 감동이네요 ~
생성문장 (부정)	영화 내용은 그냥 그렇다고 재미없어서 그냥 그냥 그냥
입력문장 (긍정)	송강호와 강동원의 오묘한 케미는 끝내준다.
생성문장 (긍정)	강지환씨의 연기력도 좋고, 감동적이었음.
생성문장 (부정)	강지환은 아무리 드라마라지만, 그래도 이건아니다.

표 5. 긍정 문장 입력에 대한 생성 문장들

입력문장 (부정)	기대를 너무했나 갑자기 여자친구를 죽이질않나 여튼 좀 아쉬운영화
생성문장 (부정)	기대안하고 봤는데 정말 재미없다 그냥 그냥 그냥그냥그냥그래
생성문장 (긍정)	기대안하고 봤는데 정말 재미있게 잘봤습니다 정말 감동적이네요
입력문장 (부정)	개봉당시엔 상당히 파격적이었겠지만 지금 보기에는 너무 진부한 고전영화.
생성문장 (부정)	개봉한지도 모르겠지만 그래도 이영화는 정말 재미없다.정말 재미없다.
생성문장 (긍정)	개봉한지도 모르지만 그래도 재미있는 스토리와 연출이 너무 좋았음.

표 6. 부정 문장 입력에 대한 생성 문장들

입력문장 (긍정)	와우~~~~엄청난영화입니바 케미는 끝내준다.
생성문장 (긍정)	와 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
생성문장 (부정)	와 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

표 7. 생성 문장이 제대로 감성을 담지 못하는 경우

[9]의 연구에서도 영화에 대한 긍부정 문장들을 VAE를 사용하여 생성하였다. 표 8은 [9]에서 제시한 생성문장의 샘플로 NSMC 학습 데이터로 생성한 문장이다. 하지만 학습에 사용한 데이터라고 할 지라도 입력 문장과 출력 문장이 서로 다른 단어들을 사용하고 같은 영화에 대한 평이라고 보기 어려웠다. 표 9는 우리의 모델로 표 8과 같은 데이터에 대해 문장을 생성한 결과이다. 표 9에서의 생성문장과 입력문장은 표 8에 비해서 비교적 비슷한 내용을 담고 있다고 보여진다. 즉, 본 논문에서 제안한 방법은 문장의 의미풀기(disentanglement)를 위한 적대적 항 없이 학습하여도 생성문장들이 비슷한 영화에 대한 평이라고 생각되는 문장들을 생성해준다.

하지만 표 7와 같이 문장 생성에서 긍정과 부정이 같은 경우가 있다. 실제로 학습 데이터를 살펴보면, “와~~”의 문구가 긍정문에도 있고 부정문에도 존재한다. “와~~”의 문구는 사람이 판단할 때도 감탄사일수도 반어법일수도 있다. 즉 전체적인 문맥상 의미를 파악을 해야만 “와~~”의미를 알 수가 있다. 즉 생성 모델이 “와~~”만을 생성하지 말고 다른 의미의 토큰들도

입력문장 (부정)	평범함속에 녹아든 평범한 일상. 조금 밋밋한게 흠.
생성문장 (부정)	저렇 게 재미있 게 보 ㄴ 영화 이 ㄴ 데 그렇 는데 이 평점 너무 낮 은 듯 하 ㅁ
생성문장 (긍정)	이게뭔 데 재미없 게 볼거리 도

표 8. [9]에서 학습 데이터로 생성한 문장 예시.

입력문장 (부정)	평범함속에 녹아든 평범한 일상. 조금 밋밋한게 흠.
생성문장 (부정)	평범한 소재로 재미없는 스토리..그냥 그냥 그렇다치고는 [UNK]
생성문장 (긍정)	평범한 소재로 재미있게 본다. 그래서 더 재미있게 본영화.

표 9. [9]에서 사용한 학습 데이터 샘플로 본 논문의 모델로 생성한 문장

입력문장 (긍정)	신선한 로맨스 저절로 웃음이난다
긍정:부정 = 100:0	신선한 소재와 스토리도 좋고,감동도 있고
긍정:부정 = 80:20	신선한 소재와 스토리도 좋고,스토리도 [UNK]
긍정:부정 = 60:40	신선한 소재와 스토리도 좋고,스토리도 [UNK]
긍정:부정 = 40:60	신선한 소재와 스토리도 탄탄하고 연기력
긍정:부정 = 20:80	신선한 소재로 이렇게 재미없는 영화는 처음이다
긍정:부정 = 0:100	신선한 소재로 이렇게 재미없는 영화는 처음이다

표 10. 스타일 속성의 잠재 표현 공간 변화의 문장 생성

생성을 해야 하는데 이에 대한 장치가 없는 문제점이 있다.

추가적으로 감성 스타일을 입력으로 줄 때 긍정/부정 이분법으로 나누지 않고 잠재 벡터를 변화시키는 방법으로 생성 데이터의 변화를 관찰 해보았다. 표 10과 같이 선형적으로 총 6단계의 변화를 주면서 문장생성을 하였다. 표 10의 샘플 결과는 긍정 스타일에 해당하는 값이 스타일의 20~40%를 차지하였을 때 부정문을 생성하게 된다. 스타일이 변화는 데이터에 따라 50%의 근방에서 다르게 일어난다.

5. 결론

본 논문에서는 비병렬 데이터, 한국어 영화 리뷰 감상평 문장들의 감성 스타일을 변화시키는 연구를 하였다. [9]와 비교해보면 문장의 의미풀기(disentanglement)을 위한 적대적 학습 항이 없이도 문장의 내용을 고려하여 문장을 잘 생성함을 볼 수 있었다. 하지만 문맥적으로 파악해야 하는 이중의 의미를 담는 문장들도(예. “와~~”) 생성하는 문제점이 존재한다.

또한 BERT을 이용하는 점에서 인코더와 디코더 또한 트랜스포머 계열[10]로 대체하면 좀 더 좋은 결과를 얻을 수 있을 것으로 생각된다.

향후 연구에서는 본 논문에서 실험한 감성 스타일 속성 외에 시제, 상품의 카테고리, 성별 등을 고려하고자 한다.

참고문헌

[1] ZHU, Jun-Yan, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. 2017. p. 2223-2232.

[2] https://github.com/e9t/nsmc

[3] YU, Lantao, et al. Seqgan: Sequence generative adversarial nets with policy gradient. In: Thirty-First AAAI Conference on Artificial Intelligence. 2017.

[4] HU, Zhiting, et al. Toward controlled generation of text. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017. p. 1587-1596.

[5] SHEN, Tianxiao, et al. Style transfer from non-parallel text by cross-alignment. In: Advances in neural information processing systems. 2017. p. 6830-6841.

[6] LOGESWARAN, Lajanugen; LEE, Honglak; BENGIO, Samy. Content preserving text generation with attribute controls. In: Advances in Neural Information Processing Systems. 2018. p. 5103-5113.

[7] SUBRAMANIAN, Sandeep, et al. Multiple-Attribute Text Style Transfer. arXiv preprint arXiv:1811.00552, 2018.

[8] DEVLIN, Jacob, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[9] 김건영, 이창기, VAE와 CNN이 결합된 모델을 이용한 한국어 문장 생성과 감성 분석. 2018 한글 및 한국어 정보처리 학술대회 논문집, pp.430~433

[10] VASWANI, Ashish, et al. Attention is all you need. In: Advances in neural information processing systems. 2017. p. 5998-6008.

[11] SUTSKEVER, Ilya; VINYALS, Oriol; LE, Quoc V. Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. 2014. p. 3104-3112.

[12] KINGMA, Diederik P.; BA, Jimmy. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[13] PRABHUMOYE, Shrimai, et al. Style transfer through back-translation. arXiv preprint arXiv:1804.09000, 2018.

[14] 임승영, 김명지, 이주열, KorQuAD: 기계독해를 위한 한국어 질의응답 데이터셋, 한국정보과학회한국정보과학회 학술발표논문집 2018.12539 - 541 (3 pages)

[15] https://korquad.github.io/