KIPS Trans. Softw. and Data Eng.
Vol.8, No.6 pp.257~264 pISSN: 2287-5905

https://doi.org/10.3745/KTSDE.2019.8.6.257

Evaluation of Sentimental Texts Automatically Generated by a Generative Adversarial Network

Cheon-Young Park† · Yong-Seok Choi†† · Kong Joo Lee†††

ABSTRACT

Recently, deep neural network based approaches have shown a good performance for various fields of natural language processing. A huge amount of training data is essential for building a deep neural network model. However, collecting a large size of training data is a costly and time-consuming job. A data augmentation is one of the solutions to this problem. The data augmentation of text data is more difficult than that of image data because texts consist of tokens with discrete values. Generative adversarial networks (GANs) are widely used for image generation. In this work, we generate sentimental texts by using one of the GANs, CS-GAN model that has a discriminator as well as a classifier. We evaluate the usefulness of generated sentimental texts according to various measurements. CS-GAN model not only can generate texts with more diversity but also can improve the performance of its classifier.

Keywords : Generative Adversarial Network (GAN), Data Augmentation, Sentimental Text, Sentiment Classifier

생성적 적대 네트워크로 자동 생성한 감성 텍스트의 성능 평가

박 천 용† · 최 용 석†† · 이 공 주†††

요 약

최근 자연언어처리 분야에서 딥러닝 모델이 좋은 성과를 보이고 있다. 이러한 딥러닝 모델의 성능을 향상시키기 위해서는 많은 양의 데이터가 필요하다. 하지만 많은 양의 데이터를 모으기 위해서는 많은 인력과 시간이 소요되기 때문에 데이터 확장을 통해 이와 같은 문제를 해소할 수 있다. 그러나 문장 데이터의 경우 이미지 데이터에 비해 데이터 변형이 어렵기 때문에 다양한 문장을 생성할 수 있는 생성 모델을 통해 문장 데이터 자동 확장을 해보고자 한다. 본 연구에서는 최근 이미지 생성 모델에서 좋은 성능을 보이고 있는 생성적 적대 신경망 중 하나인 CS-GAN을 사용하여 학습 데이터로부터 새로운 문장들을 생성해 보고 유용성을 다양한 지표로 평가하였다. 평가 결과 CS-GAN이 기존의 언어 모델을 사용할 때보다 다양한 문장을 생성할 수 있었고 생성된 문장을 감성 분류기에 학습시켰을 때 감성 분류기의 성능이 향상됨을 보였다.

키워드 : 생성적 적대 신경망, 데이터 확장, 감성 텍스트, 감성 분류기

1. 서 론

최근 딥러닝(Deep-learning) 알고리즘이 많은 분야에 적용되면서 좋은 성능을 보이고 있다. 특히 딥러닝 알고리즘은 대화 시스템, 기계번역, 문서 요약 등의 자연언어처리 전 분야에서 좋은 성과를 보이고 있다. 이러한 딥러닝 알고리즘의 성능을 향상시키기 위해서는 많은 양의 학습 데이터가 필요하다. 하지만 충분한 양의 학습 데이터를 모으기 위해서는 많은 인력과 시간이 소요된다.

기계학습에서 부족한 학습 데이터에 대한 해결책 중 하나는 데이터 확장(Data Augmentation)이다[1,2]. 데이터 확장은 보유한 데이터에 적절한 변형을 가해 새로운 데이터를 얻는 방법이다. 예를 들어 이미지 데이터 확장은 학습 데이터의 크기, 방향, 밝기 등을 변형시켜 학습 데이터의 양을 늘린다. 이를 통해 이미지 분류기는 데이터 확장으로 학습 데이터의 양을 증가시켜 분류 성능을 향상시킬 수 있다. 그러나 이미지와 달리 문장과 같은 이산(Discrete) 값을 갖는 데이터는 변형이 어렵다. 따라서 문장 데이터를 확장하기 위해서는 동의어 리스트[3]를 구축하거나, 문장을 변형할 규칙(task-specific heuristic rules)[4]을 사용해야 했다.

위와 같이 동의어 사전이나 규칙 기반의 알고리즘 없이 문장 데이터를 확장하기 위해서 언어 모델(Language model)을 사용하는 방법이 고안되었다. 하지만 언어 모델은 학습한 문장을 똑같이 만들어 내는 것이 목적이기 때문에 다양한 데이터를 기대할 수 없다. 그렇기 때문에 본 연구는 생성 모델(Generative model)을 사용하여 다양한 문장 데이터를 만들어 보고 생성된 문장을 평가해 본다.

본 연구에서는 주어진 문장으로부터 새로운 문장 집합을 생성하기 위해서 생성적 적대 네트워크(Generative adversarial

※ 이 연구는 충남대학교 학술연구비에 의해 지원되었음.
† 비 회 원 : 충남대학교 전자전파정보통신공학과 석사과정
†† 준 회 원 : 충남대학교 전자전파정보통신공학과 박사과정
††† 종신회원 : 충남대학교 전파정보통신공학과 교수
Manuscript Received : March 11, 2019
Accepted : April 2, 2019
* Corresponding Author : Kong Joo Lee(kjoolee@cnu.ac.kr)

networks)[5]를 사용한다. 생성적 적대 네트워크는 생성기와 판별기로 이루어진 모델이다. 생성적 적대 네트워크의 생성기는 실제 데이터와 유사한 데이터를 생성하는 모델이다. 판별기는 입력된 데이터가 실제 데이터일 확률을 계산해주는 모델이다. 생성기는 판별기가 계산한 확률값을 바탕으로 실제 데이터와 유사한 데이터를 만들도록 학습된다. 이러한 생성적 적대 네트워크는 이미지 생성 분야에서 좋은 성과를 보이고 있다.

그러나 생성적 적대 네트워크는 문장과 같은 이산 값을 생성하기에 부적합하다. 이미지 데이터의 경우 연속적인 값을 갖는 잠재 벡터로부터 값을 미세하게 변형 시켜 데이터를 생성할 수 있다. 그러나 문장의 경우 각 단어가 이산 값을 갖기 때문에 연속적인 값을 갖는 잠재 벡터로부터 값을 미세하게 변형하여 데이터를 생성할 수 없다. 그 이유는 단어의 경우 이미지의 픽셀 값과 달리 미세하게 변형된 값을 정의 할 수 없기 때문이다[6].

또한 생성적 적대 네트워크는 생성된 데이터에 대해 판별기가 하나의 보상값만을 제공한다. 그러나 잘 만들어진 문장을 판별하기 위해서는 전체 문장에 대한 보상값도 중요하지만 각 단어의 선택이 얼마나 잘 이루어졌는지도 고려되어야 한다. 그러므로 판별기에서 주는 하나의 보상값으로는 문장의 완성도를 판별하기에는 부족하다[7].

위의 문제를 해결하기 위해서 SeqGAN[7] 모델이 제안되었다. SeqGAN은 강화 학습의 일종인 몬테카를로 트리 검색(Monte-Calro Tree Search)을 이용한 정책 그래디언트(Policy Gradient)를 채용한다. 주어진 토큰으로부터 몬테카를로 트리 검색을 통해 가능한 문장들을 완성하고 완성된 문장에 대한 보상값을 계산한다. 그리고 SeqGAN의 생성기는 이산 값을 생성할 수 있도록 LSTM(Long Short Term Memory) 모델을 사용한다. 그 결과 SeqGAN은 학습 데이터와 유사한 문장 데이터를 생성할 수 있다.

본 연구에서는 감성인식기 구축에 필요한 학습 데이터를 생성적 적대 신경망을 이용하여 자동 확장해 본다. 감성인식기를 구축하기 위해서는 긍/부정 감성별 학습 데이터가 필요하다. SeqGAN은 자동으로 문장을 생성할 수는 있지만 긍/부정과 같은 분류 정보에 따른 문장을 생성할 수는 없다. 이와 같이 분류에 따라 다른 데이터를 생성할 수 있도록 제안된 모델 중 하나가 CS-GAN(Category Sentence GAN)[9]이다. CS-GAN은 생성기에 잠재 벡터(latent vector)와 분류 정보(category information)를 추가하여 분류에 따른 문장을 생성할 수 있다. 또한 분류기(Classifier)를 추가하여 생성된 문장이 입력된 분류 정보에 맞는지 확률값을 계산하여 생성기 학습에 사용한다. 이와 같은 방법으로 CS-GAN은 입력된 분류 정보에 따라 문장을 생성할 수 있다. 본 연구는 SeqGAN을 이용한 감성인식 학습데이터 자동 생성[8]을 확장한 연구이다.

본 연구에서는 CS-GAN을 사용하여 감성 데이터를 자동으로 생성하고 생성된 감성 문장들의 유용성을 평가해 본다. 우선 자동 생성된 감성 데이터를 이용하여 긍/부정 분류기(classifier)를 학습시켜 보고 데이터 확장을 통해 기계학습기의 성능 향상이 가능한지를 확인한다. 생성된 문장 집합의 복잡도(Perplexity), 참신성(Novelty), 다양성(Diversity)을 평가해 본다. 또한 실제 문장과 모델에 의해 자동 생성된 문장을 사람이 얼마나 구분해 낼 수 있는지를 측정해 봄으로써 자동 생성된 문장의 완성도를 평가해 본다. 긍/부정의 분류 정보에 따라 적절한 문장을 생성했는지에 대해서도 평가해 본다.

2. 관련 연구

생성적 적대 네트워크[5]($V(D,G)$)는 이미지 생성 분야에서 좋은 성능을 보이는 딥러닝 모델이다. 생성적 적대 네트워크는 생성기와 판별기로 이루어져 있다. 생성기(G)는 잠재 벡터(z)를 입력받아 실제 이미지와 유사한 이미지를 만들어 낸다. 판별기(D)는 입력된 이미지 데이터가 실제 데이터인지 생성기가 만든 가짜 데이터인지 확률값을 계산한다. 생성적 적대 네트워크의 목적 함수는 Equation (1)과 같이 생성기와 판별기의 목적 함수 합으로 표현한다.

$$\min_G \max_D V(D,G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{x \sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

생성적 적대 네트워크는 이미지와 같이 연속적인(continuous) 값을 갖는 데이터를 생성할 수 있지만 문장과 같은 이산 값(Discrete)을 갖는 데이터를 생성하기에는 적합하지 않다. 이와 같은 문제를 해결하기 위해 SeqGAN이 제안되었다[7]. SeqGAN은 이산 값을 갖는 데이터 열을 만들 수 있는 생성적 적대 네트워크이다. SeqGAN의 생성기는 재귀 신경망(Recurrent Neural Networks)으로 이산 값을 갖는 데이터 열을 만들어 낸다. 판별기는 입력된 데이터 열이 실제 데이터인지 생성기가 만든 가짜 데이터인지 확률값을 계산한다. SeqGAN은 생성기를 학습하기 위해 강화학습의 일종인 몬테카를로 트리 검색을 적용한 정책 그래디언트(Policy Gradient)를 사용한다. 강화학습을 사용하기 위해 SeqGAN에서 생성기가 이산 값을 갖는 데이터 토큰(token)을 생성하는 것을 행동(action)으로 보고, 생성기가 만든 미완성 데이터 열을 상태(State)로 정의한다. 그리고 판별기가 생성한 데이터 열을 입력하여 계산한 확률값을 보상(reward)으로 사용한다. 생성기는 데이터 토큰을 생성할 때마다 몬테카를로 트리 검색을 통해 남은 열을 샘플링(sampling)하여 완성한다. 몬테카를로 트리 검색으로 완성된 열을 판별기에 입력하여 현재 만들어진 토큰에 대한 보상값을 계산한다.

하지만 SeqGAN은 특정 분류별 문장을 생성할 수 없다. 따라서 생성기에 분류 정보를 입력하여 원하는 의미의 문장을 생성하는 CS-GAN이 제안되었다[9]. CS-GAN은 생성기, 판별기 그리고 분류기로 구성된다. CS-GAN의 생성기는 입력된 분류 정보에 따른 문장을 생성할 수 있다. 판별기는 문장 데이터를 입력받아 입력된 문장이 실제 데이터일 확률값을 계산한다. 분류기는 입력된 문장 데이터가 분류 정보에 얼마나 부합하는지 확률값을 계산한다. 생성기의 보상값은 판별기와 분류기의 확률값을 조화 평균으로 계산하여 사용한다.

일반적으로 생성적 적대 네트워크가 생성하는 문장은 짧고

다양하지 못하다는 문제가 있다[9]. 이 문제를 해결하기 위해서 Senti-GAN[10]은 각 분류 정보마다 각각의 생성기가 전담하여 문장을 만들도록 하였다. 따라서 Senti-GAN의 생성기는 해당 분류 정보에 대한 문장생성에만 집중할 수 있다. 판별기는 입력문장이 실제 문장인지 생성기가 만든 가짜 문장인지 구분할 수 있도록 학습시킨다. 긍/부정 의미가 담긴 상품평 문장 데이터에 대해 실험을 한 결과, 문장의 유창성(Fluency), 참신성(Novelty), 다양성(Diversity)이 향상됨을 보였다.

3. CS-GAN 모델

본 연구에서는 SeqGAN 모델에 분류기를 추가하여 CS-GAN을 구현하였다. 본 연구에서 사용한 CS-GAN은 Fig. 1과 같이 생성기(Generator)와 판별기(Discriminator), 분류기(Classifier)로 구성된다. 생성기는 매 단계마다 긍/부정 감성 정보(C)를 입력받아 긍/부정 의미가 담긴 문장을 무작위로 생성한다. 판별기는 입력 문장이 실제 문장 데이터(Real Data)인지 생성된 문장 데이터(Generated Data)인지 구분하고, 분류기는 입력 문장이 긍/부정 의미 중 어느 분류에 속하는지 구분한다.

CS-GAN의 생성기가 만든 문장은 몬테카를로 트리 검색을 통해 각 단계에 따른 보상값을 얻는다. 생성기가 만든 각 단어에 대한 보상값은 판별기와 분류기가 계산한 확률값을 사용한다. 그 후 판별기와 분류기로부터 얻은 보상값으로 생성기를 학습한다.

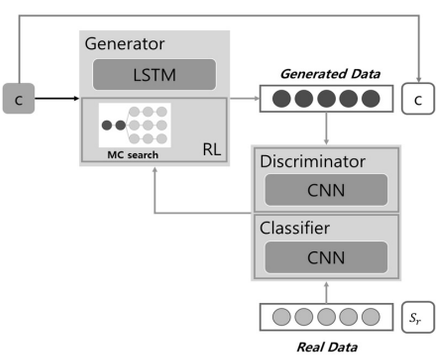


Fig. 2. The Architecture of CS-GAN

3.1 생성기

CS-GAN의 생성기는 Fig. 2와 같이 LSTM 모델을 사용한다. LSTM의 입력(x_t)은 Equation (2)와 같이 문장의 단어 임베딩(w_t)과 긍/부정 정보를 담고 있는 임베딩($\varphi(C)$)의 결합이다.

$$x_t = (w_t;\varphi(C)) \tag{2}$$

생성기의 LSTM에서 t 단계의 은닉 상태(hidden state) h_t는 전 단계 은닉 상태 h_{t-1}와 문맥 정보(Context;c_{t-1}), 입력(x_t)를 이용하여 Equation (3)과 같이 계산한다. 출력 단어($\hat{d}$)는 Equation (4)와 같이 은닉 상태에 가중치(V)와 상수(b)를 계산하여 가장 높은 확률을 갖는 단어로 선택한다.

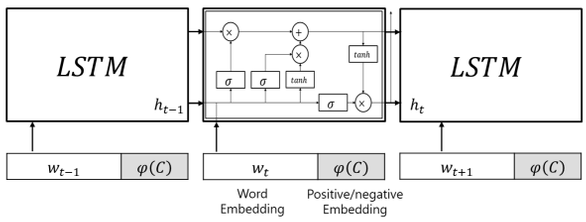


Fig. 1. The Generator of CS-GAN (w_t is the Word-embedding and $\varphi(C)$ is the Category Embedding.)

$$h_t = LSTM(h_{t-1}, C_{t-1}, x_t) \tag{3}$$

$$p(\hat{d} \mid x_1, \cdots, x_t) = z(h_t) = softmax(Vh_t + b) \tag{4}$$

3.2 판별기와 분류기

본 연구에서 판별기와 분류기는 동일한 합성곱 신경망 구조를 사용하는데 이를 Fig. 3에 도식화 하였다. 두 모듈의 차이점은 학습하게 되는 최종 출력 값인데, 판별기는 출력값이 실제(real) 데이터인지 가짜(fake) 데이터인지에 대한 확률값인데 비해 분류기는 긍정인지 부정인지에 대한 확률값이다.

합성곱 신경망의 입력($\varepsilon_{1:T}$)은 Equation (5)과 같이 문장을 이루는 단어들의 임베딩(w_t) 결합이다. Equation (6)~(8)을 이용하여 최종 출력의 확률값을 계산한다. 본 연구에서는 다양한 N-gram을 반영하기 위해서 여러 개의 필터 크기를 사용해 문장을 표현한다.

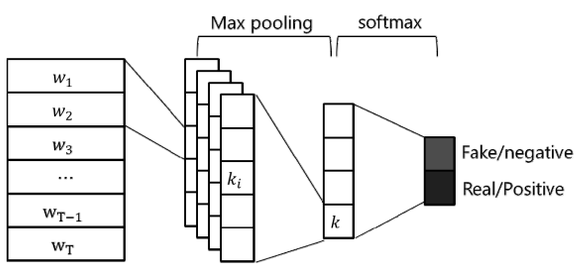


Fig. 3. The Discriminator/Classifier of CS-GAN

$$\varepsilon_{1:T} = w_1 \oplus w_2 \oplus \cdots \oplus w_T \tag{5}$$

$$k_i = \rho(v \otimes \varepsilon_{i:i+l-1} + b) \tag{6}$$

$$\tilde{k} = \max\{k_1, \cdots, k_{T-l+1}\} \tag{7}$$

$$t = sigmoid(W\tilde{k} + b) \tag{8}$$

분류기와 판별기가 출력한 확률값은 생성기가 긍/부정 감성 정보에 따라 문장을 잘 생성했는지에 대한 보상값으로 사용된다. 분류기와 판별기는 생성기가 만든 문장 데이터와 실제 데이터를 모두 사용하여 학습한다. 본 연구에서는 판별기와 분류기의 성능을 향상시키기 위해서 하이웨이 네트워크(Highway networks)[11]를 사용한다. 하이웨이 네트워크를 사용함으로써 많은 신경망 층으로 이루어진 딥러닝 모델의 학습 속도 저하를 막을 수 있다.

Algorithm

Require: generator G_{θ_g} ; policy G_γ ;discriminator D_{θ_d}; Classifier D_{θ_c} ; Dataset $S = W_{1:t}$

Initialize the parameters of G_{θ_g}, $D_{\theta_d}$and D_{θ_c}
Use the positive samples W to pre-train the generator
$\gamma \leftarrow \theta_g$
Generate the negative samples using G_γ for training D_{θ_d} and D_{θ_c}
Pre-train discriminator and classifier
repeat
  for g-steps **do**
    Generate a sequence $d_{1:t} \sim G_{\theta_g}$
    for t in 1:T **do**
      Compute $Q^{G_{\theta_g}}_{<D_{\theta_g}, D_{\theta_d}>}$ using **Eq. (10)** with policy samples
    end for
    Update parameters of generator in Eq. (11).
  end for
$\gamma \leftarrow \theta_g$
  for d-steps **do**
    Update the parameters of discriminator with the generated sentences and real sentences using Eq. (12)
  end for
  for classifier-epochs **do**
    Train the classifier with generated labeled sentences and the real labeled sentences using Eq. (13)
  end for
until model convergence

Fig. 4. The Training Algorithm of CS-GAN

3.3 알고리즘

Equation (9)는 CS-GAN 모델의 목적 함수이다. 시작 단어(w_0)와 긍/부정 감성 정보(C)가 주어졌을 때 얻을 수 있는 보상값($Q^{G_{\theta_g}}_{<D_{\theta_d}, D_{\theta_c}>}$)이 최대가 되도록 하는 것이다. θ_g,θ_d,θ_c는 각각 생성기, 판별기, 분류기의 파라미터(parameter)이고 $\hat{d}_g$는 생성기에서 만들어낸 단어이다. 그리고 D_{θ_d};D_{θ_c}는 판별기와 생성기가 계산한 확률값이다.

$$J(\theta_g) = E[R_T | w_0, C; \theta_g] = \sum_{\hat{d}_g \in W} G_{\theta_g}(\hat{d}_g | w_0, C)\, Q^{G_{\theta_g}}_{<D_{\theta_d}, D_{\theta_c}>}(w_0, \hat{d}_g) \quad (9)$$

생성기에 분류기와 판별기의 보상값($Q^{G_{\theta_g}}_{<D_{\theta_d}, D_{\theta_c}>}$)을 하나의 보상값으로 계산하기 위해 조화 평균을 사용하였다[9]. 분류기가 계산한 확률값과 판별기가 계산한 확률값을 Equation (10)과 같이 계산하여 생성기의 강화학습 보상값으로 정의한다.

$$Q^{G_{\theta_g}}_{<D_{\theta_g}, D_{\theta_d}>} = \frac{2 D_{\theta_d} D_{\theta_c}}{D_{\theta_d} + D_{\theta_c}} \quad (10)$$

Fig. 4는 모델의 학습 알고리즘이다. 먼저 생성기, 판별기, 분류기의 파라미터를 초기화한다. 실제 문장 데이터(W)로 생성기를 사전 학습(pre-training)한다. 생성기의 사전학습 후 몬테 카를로 트리 검색을 실시하는 정책 모델을 생성기의 파라미터와 같은 값으로 초기화 한다. 사전 학습한 생성기가 만든 문장 데이터와 실제 문장 데이터를 이용해 판별기, 분류기를 사전 학습한다. 사전 학습이 끝나면 생성기가 문장을 생성한다. 생성된 문장은 판별기와 분류기를 통해 보상값이 계산된다. 생성기가 $t(0 < t < T)$길이 만큼 단어를 생성하면 $t+1$길이부터 T 길이까지 정책 모델이 단어를 샘플링한다. 정책 모델이 완성한 문장을 판별기와 분류기에 입력으로 주어 보상값을 계산한다. 계산한 보상값으로 Equation (11)과 같이 생성기의 파라미터를 갱신한다. 그 후 정책 모델의 파라미터를 생성기의 파라미터로 갱신한다.

$$\theta_g \leftarrow \theta_g + \alpha_t \nabla_{\theta_g} J(\theta_g) \quad (11)$$

생성기의 학습이 끝나면 생성기는 문장 데이터를 만든다. 생성기가 만든 문장 데이터와 실제 문장 데이터를 사용하여 판별기와 분류기를 학습한다. 판별기는 Equation (12)와 같이 실제 문장과 생성기가 만든 문장을 구분하도록 학습한다. 분류기는 Equation (13)처럼 입력 문장이 긍/부정 중 어느 의미를 담고 있는지 구분하도록 학습한다. 생성기와 판별기, 분류기의 학습 과정을 반복한다.

Table 1. Information of Training Data Set

	Number of sentences	Average number of morphemes in a sentence (Positive/negative)	Average number of Eojeols in a sentence (Positive/negative)
Train data	6000	14.58/13.38	6.82/6.31
Test data	2000	14.49/13.51	6.76/6.38

Table 2. Examples of Training Data

	Examples
Positive sentences	이렇게 <SP> 각성 할 <SP> 줄 이야 <SP> 잘했 다 <EOS> 다 들 <SP> 고생 <SP> 수고 <SP> 했 어요 <EOS> 오늘 <SP> 역대 급 <SP> 골 <SP> 결정 력 이었 다 <EOS> 오늘 <SP> 경기 력 <SP> 너무 <SP> 좋 다 <EOS> 황의조 <SP> 완전 <SP> 잘한 다 <EOS>
Negative sentences	감독 <SP> 아 <SP> 진짜 <SP> 답 <SP> 없네 <EOS> 김 감독 <SP> 진짜 <SP> 새 가슴 <SP> 답답한 <SP> 인간 <EOS> 국 대 를 <SP> 시킬 <SP> 인성 이 <SP> 못 되 네요 <EOS> 손흥민 <SP> 진짜 <SP> 거품 <SP> 중 의 <SP> 거품 이다 <EOS> 인성 이 <SP> 글러 <SP> 먹은 <SP> 놈 들 <EOS>

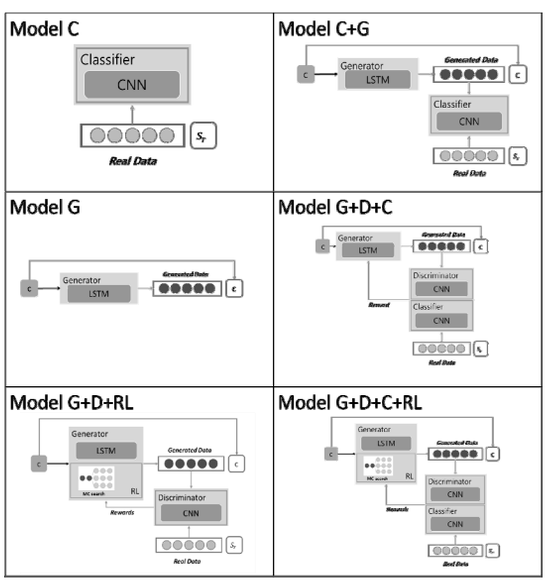


Fig. 5. The Various Models used in the Evaluation

$$L_D = H_{E_{d_g \sim (c,z)}} + H_{E_{d_h \sim data}}$$
$$H_{E_{d_g \sim (c,z)}} = E_{d_g \sim (c,z)}[-\log(1 - D(G(\varphi(c), \theta_g)), \theta_d)] \quad (12)$$
$$H_{E_{d_h \sim data}} = E_{d_r \sim data}[-\log(D(s_r, d_r, \theta_d))]$$

$$L_C = Loss(< C(d_r; \theta_c), s_r >) + Loss(< C(G(\varphi(c); \theta_g); \theta_c), c >) \quad (13)$$

4. 실험 환경

실험에 사용한 데이터는 네이버 스포츠 뉴스 중 2018 아시안 게임 축구 분야의 댓글이다. 수집한 댓글은 긍정 댓글과 부정 댓글로 나누어 각각 4,000개의 데이터를 수집하였다. 선별한 댓글 데이터는 Konlpy의 Open Korean Text[1]를 사용해 형태소 단위로 분해하고 띄어쓰기 부분은 “<SP>”토큰으로 대체하였다. 얻은 데이터의 긍/부정 댓글 각각 3,000개씩은 학습데이터로 사용하고, 나머지는 1,000개씩은 평가데이터로 사용하였다. 학습데이터의 정보는 Table 1과 같다. Table 2는 학습 데이터의 예제이다.

5. 실험 결과 및 평가

본 실험에서 비교 평가할 모델들을 Fig. 5에 도식화하였다.

모델 C는 CNN 구조로 이루어진 분류기로 실제 학습 데이터로만 학습한 분류기이다.

모델 C+G는 모델 C와 같은 구조로 모델 G가 생성한 문장까지 사용하여 학습시킨 분류기이다.

모델 G는 LSTM으로 생성된 출력이 학습 데이터의 문장과 똑같이 만들어지도록 최대 우도 예측법(Maximum Likelihood Estimation)으로 학습된 생성 모델이다.

모델 G+D+C는 CS-GAN에서 강화학습(RL)을 사용하지 않고 완성된 문장을 판별기에 입력하여 나온 확률을 생성기의 각 토큰에 대한 보상값으로 사용한 모델이다.

모델 G+D+RL은 CS-GAN에서 분류기(C)를 제거한 생성 모델이다.

모델 G+D+C+RL은 본 논문에서 제안한 CS-GAN 모델로 강화학습과 분류기를 모두 사용하는 모델이다.

Table 3는 사전 학습만 실시한 모델 G (LSTM모델)와 모델 G+D+C+RL(CS-GAN 모델)이 생성한 문장의 예시이다.

1) http://konlpy.org/ko/v0.5.1/api/konlpy.tag/#okt-class

Table 3. Examples of the Sentences Generated by LSTM and CS-GAN Models

	Model G (LSTM)	Model G+D+C+RL (CS-GAN)
Positive sentences	전경기 진짜 잇게 잘하더라 진짜 오늘 잘한 경기 다 다행인게 김민재도 잘했다 열심히 뛴 그대가 있었다	열심히 뛴 그대 선수들 고생하셨어요 힘 내서 게임 이겨서 다행 황의조 참 고맙다 오늘 간만에 멋진 골 굿 쩔어
Negative sentences	조현우가 이번에 그만두고 있어라 신태용 다시 같은 놈 들 참 노답 김민재도 좀 해봐야 밉네 근데 손흥민 존나 좀 못하는데	실력은 없고 겉멋만 들었다 의조 진심 군대 안된다 축알못 새키 또 욕 먹었다 골키퍼 다 추해요

5.1 분류기 성능 평가

Table 4는 본 논문에서 구현한 모델 중 분류기의 성능을 평가한 결과이다. 모델 C만 실제 학습 데이터로 학습한 분류기이고 나머지 모델들은 모두 학습기가 자동 생성한 문장들까지 학습시킨 분류기이다. 평가는 학습에 포함되지 않은 평가 데이터 2,000 문장에 대해 실시하였다. 모델 C+G의 경우 모델 C에 비해 2.7%의 성능 향상을 보였으며, 모델 G+D+C+RL(CS-GAN)의 경우에는 모델 C에 비해 약 3.5%의 정확도 향상을 보였다. 생성 모델을 이용하여 학습 데이터를 자동 확장하는 것이 분류기 성능을 큰 폭으로 향상시킬 수 있었다.

Table 4. Accuracy of the Classifier Models

Classifier models	Accuracy
Model C	84.0%
Model C+G	86.7%
Model G+D+C	85.5%
Model G+D+C+RL	**87.5%**

5.2 생성기 성능 평가

생성기 학습에서 어려운 문제 중 하나는 모드 붕괴(mode collapse) 현상이다[5]. 모드 붕괴는 생성기가 다양한 출력을 만들어내지 못하고 비슷한 결과만 계속해서 생성하는 경우를 뜻한다. 그렇기 때문에 생성기가 얼마나 다양한 문장을 만들어 낼 수 있는지에 대해 평가해 보고자 한다.

우선 모델이 자동 생성한 문장 중, 실제 학습 데이터에 존재하지 않으면서 서로 다른 문장의 비율을 확인해 보기 위하여 각 모델 당 100,000 문장을 생성해 보았다. 100,000 문장 중 학습 데이터에 존재하지 않으면서 고유한 문장의 비율과 이 문장들의 평균 형태소/어절 수를 Table 5에 제시하였다. 강화학습(RL)을 사용한 두 모델은 학습 데이터보다 평균 어절 수가 더 긴 문장을 생성하였다. 모델 G를 제외하면 모델 G+D+C+RL(CS-GAN)의 고유 문장 생성 비율이 가장 높다. 또한, 강화학습을 사용하지 않는 모델 G+D+C의 경우 고유 문장 생성 비율이 상당히 낮아짐을 볼 수 있다. 문장 생성 시 각 토큰만을 고려하지 않고 완성된 문장을 고려하여 보상값을 부여하는 강화학습이 새로운 문장을 생성하는데 기여함을 확인할 수 있었다.

각 모델이 얼마나 다양한 표현을 사용했는지 살펴보기 위해서 생성한 100,000 문장에 나타난 고유한 N-gram 정보를 Table 6에 제시하였다. CS-GAN 모델이 uni-gram, bi-gram, tri-gram 모두에서 고유한 표현이 제일 많이 나타났다. 또한 강화학습을 사용하는 G+D+RL과 G+D+C+RL(CS-GAN) 모델이 다른 모델들보다 훨씬 다양한 bi-gram과 tri-gram을 사용하였다. 문장 생성시 강화학습을 사용할 경우 출현 빈도가 낮은 표현들이 더 많이 선택될 수 있었다.

Table 5. The Ratio of Unique Sentences Generated by the Models

Models	Ratio of unique sentences	Average numbers (Morphemes/eojeols)
G (LSTM)	25.17	13.58 / 6.93
G+D+C	11.12	13.23 / 6.80
G+D+RL	20.88	14.37 / 7.33
G+D+C+RL (CS-GAN)	**21.08**	14.05 / 7.28

Table 6. The Number of Unique n-grams in Sentences Generated by the Models

Models	Uni-gram	Bi-gram	Tri-gram
G (LSTM)	945	3,580	13,431
G+D+C	994	3,769	14,539
G+D+RL	840	4,014	18,603
G+D+C+RL (CS-GAN)	**996**	**4,074**	**19,337**

생성 모델이 자동 생성한 문장들을 좀더 세밀히 평가해 보기 위해 문장의 참신성(novelty)과 다양성(diversity), 그리고 복잡도(perplexity)를 평가해 보았다[10]. 참신성과 다양성, 복잡도는 학습 데이터로 사용한 6,000개의 문장과 각 모델이 생성한 6,000개의 문장을 비교하여 평가한다. Table 5에서 사용한 고유한 문장 중에서 동일 단어가 3회 이상 반복되는 문장들은 제거하고 6,000개를 무작위로 추출하였다.

참신성은 모델이 학습한 문장 외에 얼마나 새로운 표현들을 생성했는지 평가하는 지표이다. 참신성은 Equation (14)와 같이 학습 데이터 문장의 단어 집합(Bag of words)와 모델이 생성한 문장의 단어 집합의 자카드 유사도(Jaccard similarity function; J)를 계산한다. 참신성의 값이 클수록 새로운 문장이 생성됨을 의미한다.

$$Novelty(G_i) = 1 - \max J(G_i, S_j)_{j=1}^{j=|S|} \quad (14)$$

다양성은 모델이 얼마나 다양한 문장을 생성했는지 평가하는 지표이다. 다양성은 Equation (15)와 같이 모델이 생성한 문장들의 단어 집합 간 자카드 유사도를 계산한다. 다양성도 그 값이 클수록 다양한 문장이 생성됨을 의미한다.

$$Diversity(G_i) = 1 - \max J(G_i, G_j)_{j=1}^{j=|G|, j \neq i} \quad (15)$$

일반적으로 언어 모델의 성능 평가로 복잡도(perplexity)를 사용한다[12]. 이는 언어 모델이 얼마나 다음 단어를 잘 예측하는지를 평가하는 방법으로 그 값이 낮을수록 잘 학습된 언어 모델이라고 할 수 있다. 본 연구에서는 복잡도로 생성된 문장의 유창성을 측정해 본다[9]. 문장의 복잡도는 언어 모델 SRILM[2]을 사용해 계산하였다. 복잡도가 낮을수록 생성된 문장의 유창성이 좋다고 할 수 있다. Table 7은 참신성, 다양성, 복잡도에 대한 평가 결과이다.

Table 7. The Evaluation Results of Sentences Generated by the Models

Models	Novelty	Diversity	Perplexity
G (LSTM)	0.74	0.37	19.53
G+D+C	0.75	0.43	22.63
G+D+RL	0.76	0.46	20.96
G+D+C+RL (CS-GAN)	0.75	0.45	20.95

모델 G는 학습데이터와 동일한 문장을 만들도록 학습된 LSTM 모델이다 보니 모델의 복잡도가 가장 낮은 결과를 보였다. Table 5에서 모델 G는 고유한 문장의 비율이 다른 모델에 비해 높았다. 그러나 Table 6과 Table 7의 결과를 함께 고려해 보았을 때, 모델 G는 빈도가 높은 n-gram만을 이용하여 문장을 생성하기 때문에 참신성과 다양성에서 다른 모델에 비해 낮은 성능을 보임을 알 수 있었다.

참신성과 다양성 면에서는 G+D+RL 모델이, 복잡도 면에서는 모델 G을 제외하고는 G+D+C+RL (CS-GAN) 모델이 가장 좋은 성능을 보였다. 문장 생성에서는 강화학습을 도입하는 것이 다양하고 새로운 문장을 생성할 수 있음을 확인하였다.

Table 8은 모델 G+D+C+RL(CS-GAN)에서 생성기와 판별기, 분류기 학습비에 따른 각 CS-GAN 모델의 참신성, 다양성, 복잡도, 분류기의 정확도를 나타낸 것이다.

Table 8. Performance of Model G+D+C+RL(CS-GAN)

Model	Learning Ratio	Novelty	Diversity	Fluency	Accuracy
G+D+C+RL (CS-GAN)	1:1	0.747	0.443	21.36	85.1
	1:3	0.751	0.446	21.96	86.9
	1:5	**0.752**	**0.450**	**20.95**	**87.5**

2) http://www.speech.sri.com/projects/srilm/download.html

실험 결과 생성기와 판별기, 분류기의 학습 비를 1:5로 하였을 때 분류기의 정확도가 가장 높았다. 따라서 생성기와 판별기, 분류기의 학습 비를 잘 조절하는 것이 모델의 성능을 향상 시키는데 영향을 주는 것을 알 수 있다.

5.3 자동 생성된 문장에 대한 정성 평가

본 장에서는 생성 모델이 생성한 문장을 사람이 직접 평가해 보는 정성 평가를 수행해 보았다. 5.2절에서 추출한 6,000 문장을 복잡도 순으로 정렬한 후 상위 10%의 문장과 하위 10%의 문장을 배제하고 나머지 80%의 문장에서 평가 문장을 추출하였다. 복잡도 상위 10%의 문장을 배제한 이유는 상위 문장의 경우 문장의 길이가 너무 짧거나 비슷한 문장이 많이 포착되었기 때문이다. 복잡도 하위 10% 문장을 배제한 이유는 문장의 형태가 온전하지 않은 경우가 많았기 때문이다. 남은 80%의 문장을 5개 부분으로 나누어 각 부분에서 20개의 문장씩 총 100문장을 무작위로 선정하였다. 긍정 문장과 부정 문장에서 각각 100문장씩 추출하여 모델 당 200문장씩 수동 평가를 실시하였다.

정성 평가는 두 가지로 수행하였다. 첫째는 평가자가 생성된 가짜 문장과 진짜 문장을 구분할 수 있는지를 측정해 보았다. 사람이 작성한 진짜 문장이라고 판단될 경우 +1점을, 기계가 생성한 가짜 문장이라고 판단될 경우 -1점을, 기계가 생성한 것인지 사람이 작성한 것인지 구별하기 어렵다면 0점을 부여하도록 하였다. 이를 문장 판별 점수(sentence distinction score)라고 하고 문장 판별 점수가 양수 1에 가까울수록 사람이 작성한 문장과 유사하다는 의미이다. 두 번째는 긍/부정 분류에 맞도록 문장 생성이 이루어졌는지에 대한 평가이다. 평가자는 주어진 문장이 긍정이면 'P'를 부정이라고 판단되면 'N'을 부여한다. 이를 기계가 문장을 생성할 때 사용했던 태그 정보와 비교해서 긍/부정 일치도를 측정하였다.

Table 9는 평가자 8명의 점수 평균을 계산한 결과이다.

CS-GAN 모델로 생성한 문장이 LSTM으로 생성한 문장에 비해 사람이 만든 문장과 유사했다. 또한 긍/부정 분류별 생성에 있어서도 높은 일치도를 보였다.

Table 9. Qualitative Evaluation Results of CS-GAN and LSTM Models

Model	Sentence Distinction Score	Agreement (Positive)	Agreement (Negative)
G (LSTM)	-0.315	78.6/100	84.6/100
G+D+C+RL (CS-GAN)	**-0.148**	**82.9/100**	**85.0/100**

6. 결 론

본 연구에서는 GAN 모델 중 하나인 CS-GAN을 통해 새로운 문장 데이터를 자동으로 생성하고 평가해 보았다.

CS-GAN 모델을 사용하여 긍/부정 감성 레이블을 갖는 문장을 자동으로 확장하고 이를 분류기의 학습데이터로 추가

사용했을 때, 분류기의 성능이 최대 3.5%까지 향상되었다. 이는 생성 모델을 이용한 학습 데이터 자동 확장이 분류기 성능 향상에 실질적 도움이 될 수 있음을 의미한다.

CS-GAN은 문장 생성에 강화학습을 사용한다. 문장 생성에서 강화학습은 더 길고 더 다양한 n-gram 표현을 갖는 문장을 생성할 수 있도록 하였다. 생성된 문장의 참신성과 다양성에서도 강화학습을 사용하는 모델들이 더 좋은 성능을 발휘하였다. 그러나 복잡도의 경우에는 MLE(maximum liklihood estimation)로 학습한 LSTM 모델이 여전히 더 낮은 복잡도를 가졌다. 이는 LSTM 모델은 학습데이터와 동일한 문장만을 생성하도록 학습되었기 때문으로 판단된다.

본 연구에서는 판별기와 분류기의 출력값을 강화학습에서 문장의 보상값으로 사용하였다. 강화학습은 보상값이 최대가 되도록 학습하기 때문에 문장에 대한 보상값을 어떻게 측정할지가 매우 중요하다. 향후 과제는 문장의 보상값과 이를 강화학습에 어떻게 반영할지에 대한 연구를 통해 생성기 모델의 성능을 향상시키는 것이 될 것이다.

References

[1] GÓMEZ-RÍOS, Anabel, “Towards Highly Accurate Coral Texture Images Classification using Deep Convolutional Neural Networks and Data Augmentatio,” *Expert Systems with Applications*, Vol.118, pp.315-328, 2019.

[2] Zhang, Xiang, Zhao, Junbo, and Lecun, Yann, “Character-Level Convolutional Networks for Text Classification,” *Advances in Neural Information Processing Systems*, pp.649-65, 2015.

[3] KOBAYASHI, Sosuke, “Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations,” *arXiv Preprint arXiv:1805.06201*, 2018.

[4] Deschacht, Koen and Marie-Francine Moens, “Semi-Supervised Semantic Role Labeling using the Latent Words Language Model,” *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, 2009.

[5] Goodfellow, Ian, et al., “Generative Adversarial Nets”, *Advances in Neural Information Processing Systems*, pp.2672-2680, 2014.

[6] Goodfellow, Ian, “Generative Adversarial Networks for Text,” http://goo.gl/Wg9DR7, 2016.

[7] YU, Lantao, et al., “Seqgan: Sequence Generative Adversarial Nets with Policy Gradient,” *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

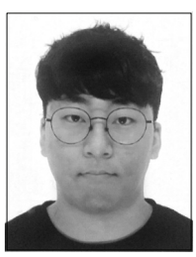
[8] C. Y. Park and K. J. Lee, “Automatic Generation of Training Corpus for a Sentiment Analysis Using a Generative Adversarial Network,” *Proceedings of the 30th Annual Conference on Human and Cognitive Language Technology*, pp.389-393, 2018.

[9] L. I. Yang, et al., “A Generative Model for Category Text Generation,” *Information Sciences*, Vol.450, pp.301-315. 2018.

[10] Wang, Ke and Wan, Xiaojun. “SentiGAN: Generating Sentimental Texts via Mixture Adversarial Networks,” *IJCAI*, pp. 4446-4452, 2018.

[11] SRIVASTAVA, Rupesh Kumar, GREFF, Klaus, SCHMIDHUBER, Jürgen, “Highway Networks,” *arXiv Preprint arXiv:1505.00387*, 2015.

[12] MIKOLOV, Tomáš, et al., “Recurrent Neural Network Based Language Model,” *Eleventh Annual Conference of the International Speech Communication Association*. 2010.



박 천 용

http://orcid.org/0000-0002-8166-2166
e-mail : sdpcy0520@gmail.com
2019년 충남대학교 전파정보통신공학과(학사)
2019년～현 재 충남대학교
전자전파정보통신공학과 석사과정
관심분야 : Natural language processing

최 용 석

https://orcid.org/0000-0002-7889-8004
e-mail : yongseok.choi.92@gmail.com
2016년 충남대학교 정보통신공학과(학사)
2018년 충남대학교
전자전파정보통신공학과(석사)
2018년～현 재 충남대학교
전자전파정보통신공학과 박사과정
관심분야 : 자연언어처리, 정보검색, 기계학습, 인공지능

이 공 주

https://orcid.org/0000-0003-0025-4230
e-mail : kjoolee@cnu.ac.kr
1992년 서강대학교 전자계산학과(학사)
1994년 한국과학기술원 전산학과(공학석사)
1998년 한국과학기술원 전산학과(공학박사)
1998년～2003년 한국마이크로소프트(유)
연구원
2003년 이화여자대학교 컴퓨터학과 대우전임강사
2004년 경인여자대학 전산정보과 전임강사
2005년～현 재 충남대학교 전파정보통신공학과 교수
관심분야: 자연언어처리, 기계학습, 인공지능, 정보검색